



Universidade do Minho

Escola de Engenharia

João Davide Barbosa Martins

**Análise de meta-modelos de escalonamento
da produção para problemas em ambientes
orientados ao produto**



Universidade do Minho

Escola de Engenharia

João Davide Barbosa Martins

**Análise de meta-modelos de escalonamento
da produção para problemas em ambientes
orientados ao produto**

Dissertação de Mestrado

Mestrado em Engenharia de Sistemas

Trabalho efetuado sob a orientação da

Professora Doutora Maria Leonilde Rocha Varela

e do

Professor Doutor Bruno Samuel Ferreira Gonçalves

Despacho RT - 31 /2019 - Anexo 3

Declaração a incluir na Tese de Doutoramento (ou equivalente) ou no trabalho de Mestrado

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos.

Assim, o presente trabalho pode ser utilizado nos termos previstos na licença abaixo indicada. Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do RepositóriUM da Universidade do Minho.

Licença concedida aos utilizadores deste trabalho



Atribuição-NãoComercial-SemDerivações

CC BY-NC-ND

<https://creativecommons.org/licenses/by-nc-nd/4.0/>

AGRADECIMENTOS

Dedico este espaço a todos aqueles que tornaram possível, de forma direta ou indiretamente a realização desta dissertação. E desde já manifesto o meu sincero agradecimento.

Em primeiro lugar queria deixar um agradecimento à Professora Doutora Maria Leonilde Rocha Varela e ao Professor Bruno Samuel Ferreira Gonçalves pelo apoio e orientação na realização desta dissertação.

Agradeço à minha família e à todos os meus colegas que me acompanharam neste longo percurso académico pelo seu apoio incondicional. Um especial apreço às minhas colegas Marta Azeredo Morais e Anaís Veloso Silva pela constante disponibilidade, apoio e motivação durante à realização deste trabalho.

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

ABSTRACT

In a market oriented to meeting consumer needs, quality standards are raised. Therefore, efficiency is one of the factors that allows us to have good results in relation to the competition. The increased need for high-volume production and the flexibility to customize products to meet consumer needs increases the importance of production scheduling. To assist the implementation of production management, new technologies are developed, giving rise to hybrids of analytical and simulation methodologies. These combined minimize the disadvantages of the separate methods, originating solutions more quickly or efficiently solutions.

In this dissertation, several line configurations of a production system are studied and analyzed using a metamodel. The methodology is applied to a product family with different levels of dissimilarity in the production sequence of the product of each product. The main focus in the approach is the simulation model, which parameter of parameters and restrictions of the scenarios is intended to verify if there are relations between the various variables of the system. For the family of products, four different configurations are studied, being the simple, parallel, reentry and skip flow shop. The implementation of the metamodel is carried out in Simio and is aided by the use of UQLab to facilitate the analysis of the results.

RESUMO

Num mercado orientado a satisfazer as necessidades do consumidor, os padrões de qualidade aumentam. Assim sendo a eficiência é um dos fatores que permite ter bons resultados face à concorrência. O aumento da necessidade de produção em altas quantidades e a flexibilidade personalização dos produtos para satisfazer as preferências do consumidor eleva a importância do planeamento da produção. Para auxiliar a implementação da gestão da produção são constantemente desenvolvidas novas tecnologias, dando origem a híbridos de metodologias analíticas e simulação. Estes métodos que combinam as duas abordagens conseguem minimizar as desvantagens dos métodos por separado originando mais rapidamente ou eficientemente soluções.

Nesta dissertação são estudadas e analisadas várias configurações de linha de um sistema de produção através de um metamodelo. Esta metodologia foi aplicada a uma família de produtos com vários níveis de dissimilitude da sequência de produção de cada produto. O principal foco na abordagem é o modelo de simulação, do qual considerando os parâmetros e restrições dos cenários analisados é pretendido verificar se existem relações entre as várias variáveis do sistema. São estudadas quatro configurações distintas sendo a linha simples, paralela, reentry e skip. A implementação do metamodelo é realizada em Símio e tem como auxílio o uso do UQLab para facilitar a análise dos resultados.

CONTEÚDO

| | | |
|-------|---|----|
| 1 | INTRODUÇÃO | 1 |
| 1.1 | Enquadramento | 2 |
| 1.2 | Objetivos | 4 |
| 1.3 | Abordagem Metodológica | 5 |
| 1.4 | Estrutura da Dissertação | 6 |
| 2 | REVISÃO DA LITERATURA | 7 |
| 2.1 | Escalonamento da Produção | 8 |
| 2.1.1 | Definição de Linhas | 9 |
| 2.1.2 | Configuração de Linhas | 10 |
| 2.1.3 | Medidas de Desempenho | 12 |
| 2.1.4 | Regras de Despacho | 12 |
| 2.2 | Simulação | 13 |
| 2.2.1 | Software de Simulação | 16 |
| 2.3 | Metamodelos | 17 |
| 2.4 | Análise Crítica da Literatura | 18 |
| 3 | METODOLOGIA FUNDAMENTADA EM METAMODELOS | 20 |
| 3.1 | Descrição do Problema | 21 |
| 3.2 | Definição dos Experimentos | 22 |
| 3.3 | Metamodelo | 22 |
| 3.3.1 | <i>Kriging</i> | 23 |
| 3.3.2 | Predições por Interpolação | 24 |
| 3.3.3 | Predições Através de Regressão | 26 |
| 3.3.4 | Trend | 28 |
| 3.3.5 | Função de Correlação | 29 |
| 4 | DESENVOLVIMENTO DO MODELO | 31 |
| 4.1 | Defenitions | 32 |
| 4.1.1 | States | 33 |
| 4.1.2 | Properties | 34 |

| | | |
|-------|---|----|
| 4.1.3 | Elements | 35 |
| 4.2 | Objetos | 36 |
| 4.2.1 | Entidades | 36 |
| 4.2.2 | Source e Sink | 39 |
| 4.2.3 | Pool | 41 |
| 4.2.4 | Workstation | 43 |
| 4.3 | Processos | 45 |
| 4.3.1 | Controle Pool | 46 |
| 4.3.2 | Seleção da Regra de Despacho | 48 |
| 4.3.3 | Seleção da Sequeuncia | 49 |
| 4.3.4 | Set Up | 50 |
| 4.3.5 | Cálculo de Estatísticas | 51 |
| 4.4 | Formulação das Regras de Despacho | 52 |
| 4.5 | Verificação dos Componentes | 53 |
| 4.6 | Análise de Precisão | 54 |
| 4.7 | Implementação do Metamodelo | 55 |
| 4.7.1 | Matérn | 56 |
| 4.7.2 | Maximum-likelihood | 57 |
| 4.7.3 | Metodo de Otimização | 58 |
| 5 | DISCUSSÃO DE RESULTADOS | 59 |
| 5.1 | Tempo na Pool | 60 |
| 5.2 | Lateness | 60 |
| 5.3 | Regras de Despacho | 60 |
| 5.4 | Configurações de Linha | 61 |
| 5.5 | Tempo de Simulação | 61 |
| 6 | CONCLUSÃO E TRABALHO FUTURO | 62 |
| | BIBLIOGRAFIA | 64 |
| A | REGRA EARLIEST DUE DATE COM 5 MINUTOS NA POOL | 74 |
| B | REGRA EARLIEST DUE DATE COM 10 MINUTOS NA POOL | 77 |
| C | REGRA EARLIEST DUE DATE COM 15 MINUTOS NA POOL | 80 |
| D | REGRA SHORTEST TOTAL PROCESSING TIME COM 5 MINUTOS NA POOL | 83 |
| E | REGRA SHORTEST TOTAL PROCESSING TIME COM 10 MINUTOS NA POOL | 86 |

| | | |
|---|--|-----|
| F | REGRA SHORTEST TOTAL PROCESSING TIME COM 15 MINUTOS NA POOL | 89 |
| G | REGRA LONGEST PROCESS SEQUENCE COM 5 MINUTOS NA POOL | 92 |
| H | REGRA LONGEST PROCESS SEQUENCE COM 10 MINUTOS NA POOL | 95 |
| I | REGRA LONGEST PROCESS SEQUENCE COM 15 MINUTOS NA POOL | 98 |
| J | REGRA LONGEST PROCESSING TIME COM 5 MINUTOS NA POOL | 101 |
| K | REGRA SHORTEST PROCESSING TIME COM 10 MINUTOS NA POOL | 104 |
| L | REGRA SHORTEST PROCESSING TIME COM 15 MINUTOS NA POOL | 107 |
| M | VARIAÇÃO DO TEMPO DE SIMULAÇÃO COM A REGRA EARLIEST DUE DATE | 110 |
| N | VARIAÇÃO DO TEMPO DE SIMULAÇÃO COM A REGRA SHORTEST TOTAL PROCESSING TIME | 113 |
| O | VARIAÇÃO DO TEMPO DE SIMULAÇÃO COM A REGRA LONGEST PROCESS SEQUENCE | 117 |
| P | VARIAÇÃO DO TEMPO DE SIMULAÇÃO COM A REGRA SHORTEST PROCES- SING TIME | 120 |

LISTA DE FIGURAS

| | | |
|-----------|---|----|
| Figura 1 | Distribuição dos centros de trabalho | 32 |
| Figura 2 | States definidos no Model | 33 |
| Figura 3 | States definidos no ModelEntity | 33 |
| Figura 4 | Exemplo de state definido como escalar | 34 |
| Figura 5 | Exemplo de state definido como vetor | 34 |
| Figura 6 | Conjunto de propriedades | 34 |
| Figura 7 | Conjunto de elementos | 35 |
| Figura 8 | Entidades | 36 |
| Figura 9 | Tabela entidades do modelo | 37 |
| Figura 10 | Tabela de sequências da entidade 1 | 38 |
| Figura 11 | Tabela Indicador | 38 |
| Figura 12 | Exemplo da definição de um source | 39 |
| Figura 13 | Atribuição da data de entrega | 40 |
| Figura 14 | Atribuição da prioridade e index de set up | 40 |
| Figura 15 | Definições de seleção da prioridade | 41 |
| Figura 16 | Tempo de entrada no sistema | 41 |
| Figura 17 | Sequenciamento | 42 |
| Figura 18 | Centros de trabalho paralelos | 42 |
| Figura 19 | Definição de um centros de trabalho | 43 |
| Figura 20 | Exemplo de tabela Set Up | 44 |
| Figura 21 | Tabela de expressões dos tempos de processamento | 45 |
| Figura 22 | Processo TimeDef | 46 |
| Figura 23 | Controle do tempo na pool | 46 |
| Figura 24 | Controle do destino dos produtos na pool das maquinas paralelas | 47 |
| Figura 25 | Seleção da regra de despacho | 48 |
| Figura 26 | Processo de seleção da sequência | 49 |

| | | |
|-----------|---|----|
| Figura 27 | Identificação da ultima entidade processada | 50 |
| Figura 28 | Calculo de estatísticas | 51 |

LISTA DE TABELAS

| | | |
|----------|---|----|
| Tabela 1 | Divisão de metodologias segundo Figueira e Lobo (?) | 8 |
| Tabela 2 | Lista de regras de despacho | 14 |
| Tabela 3 | Sequência de processamento dos produtos | 31 |

INTRODUÇÃO

As indústrias são confrontadas com a intensa competitividade global e os avanços nos campos da informação e tecnologia. O processo de produção, a qualidade e o serviço são algumas das maiores preocupações das indústrias uma vez que exercem uma grande influência nos níveis de rendimento e desempenho das mesmas. Para tal, um fator chave é a flexibilidade, de modo a assegurar a qualidade do produto e satisfazer as necessidades dos consumidores. São ainda necessárias rápidas respostas às mudanças tecnológicas, às flutuações na demanda e às mudanças de desenho nos produtos e/ou sistemas de produção. Por isso, organizações incapazes de rever as suas estratégias e adaptar os processos organizacionais correm o risco de perder as suas vantagens competitivas (Terzi and Cavalieri, 2004). Este facto é ainda mais relevante considerando que com a indústria 4.0 nos dirigimos a um futuro em que os sistemas de produção têm a necessidade de produzir produtos altamente customizados em menores quantidades, com custos reduzidos e elevada qualidade num espaço de tempo relativamente curto. Enfrentar estes desafios só é possível quando os sistemas de produção são corretamente concebidos e rapidamente modificáveis. É necessário assegurar que um sistema de produção se adapte corretamente às especificações de produção de cada produto, já existente ou novo. Aproximadamente 75% dos custos de produção de um produto são determinados no fim da escolha das especificações de produção (Ullman, 2017). O planeamento é fundamental para se assegurar que estes sistemas funcionam corretamente e se aproximam do seu máximo potencial (Yan and Wang, 2007).

O escalonamento da produção em *Job Shop* é difícil de realizar na prática, assim como na teoria, devido ao grande número de variáveis que são necessárias considerar. É

possível implementar diversas metodologias de controlo de produção para alcançar um apropriado escalonamento da produção, para que a ordem seja completa eficientemente, maximizando o uso eficaz dos recursos limitados disponíveis e minimizando o tempo de espera (Vinod and Sridharan, 2008). Entre estas metodologias estão as técnicas analíticas, meta heurísticas, abordagens baseadas em regras e simulação. No entanto, as técnicas analíticas e modelos matemáticos simples são inadequados para a análise de sistemas complexos de produção. Além disso, é frequente que os modelos matemáticos utilizem técnicas de programação matemática não adequadas para resolução de problemas complexos de planeamento de produção. A simulação é umas das melhores ferramentas para avaliar a performance de diversas políticas de escalonamento (Boris et al., 2009). Porém, esta representa, geralmente, um método demorado. Assim, e de forma a combater estas desvantagens, surge o estudo e desenvolvimento de metamodelos. Um metamodelo ajuda na optimização da simulação, fornecendo objetivos determinísticos com tempos de processamento mais reduzidos que a simulação original. A aplicação de metamodelos na gestão da produção é um tema de estudo que é necessário aprofundar, sendo as implicações e resultados da sua implementação um tema objeto de grandes estudos.

1.1 ENQUADRAMENTO

A estrutura da linha de produção é desenhada para um conjunto de produtos relacionados (família de produtos). Os problemas práticos da configuração dos sistemas de produção ocorrem, frequentemente, aquando da extensão das capacidades de produção, da introdução de novos produtos ou da modificação dos recursos disponíveis. Sendo as últimas duas situações mais comuns, a competitividade do mercado e a mudança das exigências dos consumidores requerem uma contínua inovação, novas tecnologias, processos e produtos. À medida que os produtos e as suas características são modificados, é necessário adaptar a linha de produção. Um problema muito comum na gestão desta adaptação é o facto de a estrutura global da linha de produção ser lentamente degradada. Este problema deriva das mudanças individuais numa seção da linha não se traduzirem necessariamente na melhor performance para a linha como um todo. Assim, gerir a estrutura da linha de produção é um fator critico para qualquer organização, sendo necessário assegurar a sua qualidade ao longo do tempo. Porém,

cumprir este objetivo é difícil (Critchlow et al., 2003). Os sistemas de produção flexíveis e reconfiguráveis oferecem uma forma eficiente de lidar com mudanças físicas e com a introdução de novas restrições, é imprescindível aplicá-los corretamente para aproveitar a suas vantagens. Além de considerar os diversos sistemas de produção, a fábrica precisa de utilizar eficazmente as abordagens flexíveis. Estas necessidades levam a problemas na alocação do equipamento no espaço disponível. Note-se que, a necessidade de que a disposição dos sistemas de produção corresponda aos requisitos como, por exemplo, requisitos relacionados com os recursos disponíveis, a utilização máxima das máquinas e a minimização do WIP, são algumas das restrições frequentes. Ao mesmo tempo que consideramos as diversas restrições, os cálculos na avaliação dos sistemas de produção não podem ser feitos considerando parâmetros ideais, mas uma boa solução tem que ser capaz de obter uma boa performance em ambientes dinâmicos com eventos aleatórios e parâmetros estocásticos (Gyulai et al., 2016).

A simulação demonstrou ser uma boa abordagem para avaliar sistemas complexos (Azadivar, 1992). (Carson and Maria, 1997) definem a otimização por simulação como o processo de encontrar os melhores valores das variáveis de entrada entre as combinações possíveis sem a necessidade de avaliar cada possibilidade. Por outro lado, para (Fu, 2001), é a otimização das medidas de performance com base nos resultados de simulações estocásticas. Com a simulação é possível analisar um longo período da vida real num intervalo de tempo relativamente curto. Além disso, a simulação proporciona resultados de possíveis configurações sem alterar a configuração física e número dos recursos existentes (Song et al., 2010). No entanto, apesar do poder de computação para executar simulações ser maior que há alguns anos, a complexidade dos modelos e sistemas representados também aumentou, o que torna difícil realizar análises de sensibilidade ou otimização, implicando a necessidade de uma extensiva modelagem e análise para assegurar o desenho correto do sistema de produção. Em contrapartida, a alta competitividade dos negócios e mercado impulsiona as empresas a fazer um esforço contínuo para adotar abordagens mais efetivas e eficientes, assim como ferramentas que permitam avaliar o impacto das decisões de desenho dos sistemas de produção e o seu escalonamento de produção (Zhang et al., 2019). Uma abordagem para lidar com as restrições de tempo e recursos é a utilização de metamodelos.

É possível aplicar um metamodelo através de aproximações locais, o que implica o ajuste de vários metamodelos ao longo da otimização, ou uma aproximação global, utilizando vários métodos para ajustar um metamodelo a simulação (Barton, 2009). Desta forma o tempo de execução é reduzido e/ou obtém-se uma melhor compressão do sistema simulado. Especificamente, um metamodelo pode ser usado para realizar (1) aproximações do modelo, (2) exploração do modelo, (3) formulação do problema e (4) otimização global ou com múltiplos objetivos (Wang and Shan, 2007; Yu and Popplewell, 1994). Porém, os estudos que comparam as metodologias apropriadas para a simulação são escassos (De la Fuente and Smith III, 2017).

Para a abordagem definida nesta dissertação a dificuldade de implementação do planeamento da produção é atenuada. Assim, o número de variáveis considerado é limitado, uma vez que ao considerar um maior número de variáveis tal implicaria um aumento no tempo necessário para a implementação do modelo e análise de resultados. Esta abordagem proporciona um maior controlo do sistema e facilita a identificação da relação entre as variáveis e uma extensa análise das mesmas. Na elaboração do metamodelo a compreensão das variáveis do sistema é feita com muito cuidado, de modo a obter uma representação do sistema de produção com elevada fidelidade. Alguns termos como metamodelo, família de produtos, centro de trabalho, pool e set up time são bastante utilizados, sendo a análise das relações das variáveis e desempenho do metamodelo um ponto muito importante. A presente dissertação descreve a aplicação do metamodelo com auxílio do modelo de simulação, analisando as variáveis e as relações entre as mesmas.

1.2 OBJETIVOS

O planeamento de produção recorrendo à simulação é um tema que tem recebido bastante atenção na literatura devido às suas vantagens perante outros métodos como, por exemplo, as abordagens analíticas. Apesar destas vantagens, é necessário abordar soluções que complementem a simulação de modo a reduzir as suas desvantagens. Este é um ponto cada vez mais relevante com a chegada de novos paradigmas de produção, entre eles a Indústria 4.0.

Assim, os principais objetivos desta dissertação são a investigação e aplicação de métodos de meta modelagem em simulação, melhorando o planeamento de produção de sistemas fornecendo extrapolações dos resultados da simulação e facilitando a compressão do comportamento do sistema em função das suas variáveis de entrada. É de notar que o estudo ou abordagens a estes objetivos apresentam maior importância na análise comparativa das diversas variáveis. O processo consiste em analisar as várias variáveis e comparar os valores das várias configurações de linha propostas, comprovando por último a existência de relações e implicações entre as variáveis consideradas.

1.3 ABORDAGEM METODOLÓGICA

Este projeto tem como foco estudar as relações entre as variáveis do problema do planeamento de produção, num tempo de análise relativamente reduzido queremos lidar com sistemas complexos.

Modelos matemáticos embora de rápida execução não são capazes de tratar a incerteza. Os modelos de simulação conseguem lidar com incertezas, porem são dispendiosos de executar para sistemas complexos. Analisando os dois métodos podemos então afirmar que as vantagens e desvantagens do método de otimização servem como soluções às incapacidades do método de simulação e vice-versa.

Para abordar o planeamento de produção foi escolhida uma abordagem baseada na meta modelagem. Com base em (Kianifar and Campean, 2020) a metodologia proposta é categorizada como *Kriging metamodel*, como é apresentado no capítulo 3. Este tipo de abordagem consiste em substituir o output do modelo de simulação por uma função que aproxima estes valores.

Incorporar uma lógica com conhecimento limitado da mesma é uma tarefa bastante desafiadora. Para executar e realizar uma correta implementação da metodologia existem programas especializados que facilitam este processo. Assim, após compreender a metodologia e possível uma rápida implementação da mesma. Os metamodelos são implementados em UQLab, brevemente exposto no capítulo 4.

A análise realizada nesta dissertação aborda um modelo criado para este trabalho, definido pelas necessidades de produção de uma família de produtos. No modelo apresentado cada tipo de produto seguem uma rota particular e apresentam tempos de processamento e custos específicos.

Este estudo impulsiona o aumento na utilização de estas e semelhantes metodologias. Configuração de linha, tempo de *set up*, tempo na pool, regras de despacho e tempo de simulação são alguns dos fatores críticos em estudo.

1.4 ESTRUTURA DA DISSERTAÇÃO

A dissertação consta de 6 capítulos. O presente capítulo apresenta o contexto, objetivos a alcançar, enquadramento, metodologia implementada e motivação. Seque-se a revisão da literatura onde são introduzidas as técnicas existentes para resolver o problema de planeamento de produção abordado. As descrições das metodologias de estruturação e ferramentas de apoio para este tipo de problema são apresentadas. A exposição do caso de estudo é feita no capítulo 3, assim como a teoria que suporta a metodologia utilizada. No capítulo 4 aborda-se o desenvolvimento do modelo de simulação e os passos necessários para a implementação dos metamodelos. A análise dos metamodelos obtidos são apresentados no capítulo 5. Por último, este trabalho encerra com a exposição das conclusões e trabalhos futuros que possam ser aplicados ou complementar à proposta e problema estudado.

REVISÃO DA LITERATURA

Neste capítulo é feita uma revisão a trabalhos relacionados com o tema em estudo, sendo o resultado de uma pesquisa utilizando o Google Scholar e a uma série de palavras-chave. Numa fase inicial foram encontrados documentos sobre os fundamentos da metodologia aplicada e conhecimento geral, seguindo-se uma análise mais detalhada e crítica, onde foram selecionados os documentos relacionados e com maior relevância para a área em estudo.

São inúmeros os métodos onde são combinadas a simulação e otimização, a classificação e análise das vantagens e desvantagens destes métodos já foi abordada por vários autores. Os mesmos podem ser classificados de acordo com diversos critérios, sendo um exemplo encontrado em (Carson and Maria, 1997), em que os autores recorrem à aplicação do método ou em (Fu, 2001), no qual são considerados critérios como as propriedades de otimização, a capacidade do método garantir a solução ótima dadas uma certa condição, a probabilidade do método proporcionar uma solução correta ou o fato de possuir uma heurística robusta como base. Em (Tekin and Sabuncuoglu, 2004) o foco é o problema de otimização utilizado os espaços de soluções e das variáveis para classificar os métodos em otimização local discreta, otimização local continua e otimização global. (Ammeri et al., 2011) apresentam uma classificação utilizando os mesmos critérios. É proposto por (Shanthikumar and Sargent, 1983) um esquema de classificação de acordo a utilização dos modelos de simulação e otimização do método. Entre estes estudos, um documento de elevada importância para a estruturação destas metodologias e que proporciona uma forma de localizar metamodelos no contexto geral da otimização por meio de simulação é (Figueira and Almada-Lobo, 2014) contendo

Tabela 1: Divisão de metodologias segundo Figueira e Lobo (Figueira and Almada-Lobo, 2014)

| Solution Evaluation Approaches | |
|--|--|
| . Statistical Selection Methods (SSM) | . Metaheuristics (MH) |
| . Memory-based Metaheuristics (MMH) | . Random Search (RS) |
| . Stochastic Approximation (SA) | . Sample-Path Optimization (SPO) |
| . Matemodel-based Methods | . Gradient Surface Methods (GSM) |
| . Surrogate Management Framework (SMF) | . Reverse Simulation Technique (RST) |
| . Approximate Dynamic Programming (ADP) | . Retrospective Simulation Response Optimization (RSRO) |
| Solution Generation Approaches | |
| . Solution Completion by Simulation (SCS) | . Iterative Optimization-based Simulation (IOS) |
| Analytical Model Enhancement Approaches | |
| . Stochastic Programming Deterministic Equivalent (SPDE) | . Recursive Optimization-Simulation Approach (ROSA) |
| . Function Estimation based Approach (FEA) | . Optimization-based Simulation with Iterative Refinement (OSIR) |

uma análise detalhada e crítica sobre a classificação destas metodologias. A classificação esta representada na tabela seguinte.

Da mesma forma, é possível classificar os métodos de metamodelagem seguindo diversos critérios, como, por exemplo, a classificação de metamodelos em processos Gaussianos e processos não Gaussianos de (De la Fuente and Smith III, 2017). Porém uma das mais simples e recorrentes classificações é a descrita em (Barton and Mckesheimer, 2006), sendo os mesmos critérios utilizados em (Tekin and Sabuncuoglu, 2004) e (Ammeri et al., 2011), onde os metamodelos são classificados em estratégias locais ou globais. Ao seguir estas classificações é possível observar como se encontra o estado da área da otimização da simulação e como os metamodelo estão relacionados com a área.

2.1 ESCALONAMENTO DA PRODUÇÃO

O escalonamento da produção é um dos problemas críticos em sistemas de produção, sendo extensivamente abordado em estudos na área da produção (Xia and Wu, 2005). Os sistemas de planeamento e controlo da produção estão presentes na maior parte dos sistemas de produção, sendo aplicados com o fim de proporcionar informação para gerir os recursos dos de um determinado sistema de forma a auxiliar na tomada de decisões. Estes recursos podem ser fluxos de materiais, pessoas, equipamentos, atividades internas ou fluxos de informação através do sistema. A situação em que um produto necessita ser processado em um posto de trabalho, e devido a uma ou mais restrições a sua sequência é importante, é um exemplo simples de problema de escalonamento (Heath et al., 2013). As restrições que determinam ou influenciam o planeamento da produção podem ter diversas origens, como fatores tecnológicos ou económicos. Depois de satisfazer as restrições ainda existem diversas opções possíveis para o sistema, sendo necessário selecionar uma das alternativas (Gapp et al., 1965). Feita a seleção é necessário fazer uma avaliação periódica do planeamento da produção para o atualizar e adaptar a evolução do sistema de produção, de forma a evitar diminuir a eficiência.

Um dos mais recorrentes problemas de escalonamento é o caso do *job shop*, sendo *NP-hard* (Gonçalves et al., 2005). Implicando que mesmo para instâncias pequenas do problema, não é granizada uma solução ótima. Em *job shop* existe um número finito de produtos para serem processados num número finito de máquinas, cada produto possui um conjunto de tarefas realizadas em máquinas diferentes e com tempos de processamento específicos numa ordem dependente do produto. A sequência de processamento de cada produto pode ser distinta. O *job shop* flexível é uma extensão do *job shop* sendo também um problema *NP-hard* (Pezzella et al., 2008). No *job shop* flexível é possível que uma operação de um produto possa ser realizada por qualquer máquina de um conjunto de máquinas compatíveis com a operação. Assim não só é necessário lidar com o sequenciamento de trabalhos, mas também com a designação de trabalhos as máquinas compatíveis. Isto proporciona maior flexibilidade ao sistema, assim como aumenta a dificuldade do escalonamento (Chaudhry and Khan, 2016).

2.1.1 Definição de Linhas

Em adição aos problemas de escalonamento base, foram considerados no modelo de simulação quatros conjuntos de restrições. Que através de uma heurística simples originaram quatro configurações distintas do sistema de produção. São estas configurações que são comparadas através do metamodelo, traduzindo-se na análise do desempenho de quatros configurações de linha distintas ao lidar com a mesma família de produtos.

- Linha simples, nesta linha os produtos passam em todas as máquinas uma única vez, ou seja, não é possível saltar ou visitar uma máquina. Se um produto não é processado numa máquina o seu tempo de processamento nessa máquina é 0.
- Linha com reentrada, nesta variação os produtos podem visitar máquinas. No entanto, não é permitido saltar máquinas. Da mesma forma que a linha simples se um produto não é processado num posto de trabalho o seu tempo de processamento é 0.
- Linha com salto, para esta configuração é possível saltar as máquinas onde o produto não é processado, porem não é possível visitar máquinas.
- Linha paralela, nesta linha é possível saltar e visitar máquinas. Ainda é possível a ocorrência de máquinas paralelas, o que não é possível nas configurações anteriores.

No problema abordado não é só necessário ter atenção a implementação do metamodelo, mas também na implementação do modelo de simulação e definição dos experimentos realizados. Estes componentes estão interligados, e assim sendo, um erro na implementação do modelo de simulação origina erros nos resultados e uma análise invalida por meio do metamodelo. Da mesma forma se for feita uma incorreta implementação do metamodelo, este não vai representar o modelo de simulação obtendo-se uma análise invalida do modelo e relações entre as variáveis. Desta forma a implementação do modelo começa por definir as restrições, medidas de desempenho, sequencias de processamento, tempos do sistema e todos os detalhes necessários para definir o foco do estudo. Determinados todos estes aspetos e realizada a implementação do mesmo, com a conseguinte escolha dos cenários a implementar para poder seguir

com a implementação dos experimentos. Realizados os passos anteriores o metamodelo escolhido implementado e análise dos resultados é realizada. Cada um destes aspetos é abordado com mais detalhe em posteriores secções.

2.1.2 Configuração de Linhas

Uma linha clássica ou simples é o conjunto de duas, ou mais, máquinas que realizam um determinado processamento (Garey et al., 1976b). Nestas configurações de linha não existem instâncias de máquinas que realizem o mesmo tipo de processamento ou o mesmo trabalho múltiplas vezes, significando que todas as máquinas do sistema realizam operações distintas. Além disso, na linha simples o fluxo de trabalhos é direto (Ruiz and Vázquez Rodríguez, 2010), os trabalhos com múltiplas operações seguem a mesma sequência de processamento.

Uma das definições simples da configuração da linha com máquinas paralelas ou *Parallel FlowShop* é a iteração de várias linhas Simples, ou seja, o sistema em vez de conter somente uma configuração de linha simples contém várias iterações da mesma linha simples. Resultando num sistema composto por várias linhas simples em paralelo (Al-Salem, 2004).

Pode-se considerar o *Hybrid Flow Shop* como uma generalização do problema de escalonamento no âmbito do *Parallel Flow Shop* onde as diversas linhas que compunham o *Parallel FlowShop* unem-se num único processamento. Podemos assim de outra forma considerar o *Hybrid Flow Shop* é como uma extensão da linha Simples onde existem pelo menos duas máquinas que realizam o mesmo tipo de processamento. E da mesma forma que a linha simples o fluxo de trabalhos é direto (Ruiz and Vázquez Rodríguez, 2010).

Outra derivação da linha simples é a linha com avanços de postos de trabalho ou *skip flow shop*, tratando-se de uma linha simples onde o fluxo de trabalhos não é direto. Os trabalhos não precisam de ser processados em todas as máquinas do sistema de produção.

A linha com reentradas ou *re-entry flow shop* é uma derivação da linha simples onde é possível um trabalho pode voltar a ser processado numa máquina que já tem há visitado. Geralmente ocorre quando se deteta um defeito e o trabalho tem de ser reparado.

Se o número de máquinas do sistema de produção for superior a duas o problema de minimizar o makespan é fortemente *NP-hard*, onde para número inferior de máquinas tem solução polinomial (Garey et al., 1976b). Para o caso do Hybrid Flow Shop já foi comprovado que também se trata de um problema *NP-hard* (Cheng et al., 2001). No geral os problemas reais de escalonamento da produção são *NP-hard* (Al-Salem, 2004). No entanto, se existir reentrada de trabalhos no sistema o problema de minimizar o makespan é fortemente *NP-hard* para uma configuração com duas ou mais máquinas (Pan and Chen, 2003).

2.1.3 Medidas de Desempenho

As medidas de desempenho são o meio pelo qual o desempenho dos distintos cenários é representado. Quantifica as ações que ocorrem no modelo, derivadas das diversas escolhas realizadas. Dependendo das necessidades é possível implementar diversas medidas de desempenho, no caso de este trabalho são consideradas cinco medidas de desempenho.

- *Lateness*, esta é a diferença entre a data de finalização e prazo de entrega de um produto.
- Percentagem de trabalhos finalizados no prazo de entrega.
- Número de trabalhos finalizados.
- Media de utilização dos centros de trabalho. O ideal é analisar as máquinas do sistema individualmente o que aumenta enormemente a escala de análise ao ser necessário considerar 33 *outputs* distintos em lugar das quatro medias, o que não se justifica uma vez que não estamos somente estudar este aspeto do sistema.
- Tempo no sistema, iniciando-se após os trabalhos serem libertados pela pool.

2.1.4 Regras de Despacho

Um dos métodos para incrementar a produção é a implementação de um adequado escalonamento da produção, ou mais especificamente, a definição da ordem em que os trabalhos devem iniciar a produção no espaço fabril ou em que centros de trabalho devem ser processados são as regras de despacho. Com a sua implementação é maximizada a utilização dos recursos, minimizar tempos de espera e os prazos de entrega são cumpridos (Vinod and Sridharan, 2008). Visto que para problemas de escalonamento complexos uma das melhores abordagens é procurar heurísticas que garantam resultados quase ótimos (Garey et al., 1976a).

O método de sequenciamento refere-se as regras de prioridade utilizadas para sequenciar os trabalhos, as regras de despacho são um caso especial das regras de prioridade (Haupt, 1989). Onde uma regra de despacho é a utilização de uma ou mais regras de prioridade e regras heurísticas (Gere, 1966). O *job shop* ao ser um problema computacionalmente intensivo para se resolver, resulta na frequente utilização e desenvolvimento de heurísticas e regras de despacho eficientes com tempos de processamento razoáveis (Kapanoglu and Alikalfa, 2011; Rajabinasab and Mansour, 2011). As heurísticas fundamentadas nas regras de despacho são de fácil implementação, de eficiente computação e robustas no que diz respeito a sua incerteza e variabilidade. Sendo adequadas para realizar o escalonamento em ambientes complexos, dinâmicos e de grande escala (Qiu and Lau, 2013). Ao implementar uma regra de despacho é atribuído um valor de prioridade aos trabalhos em espera com base na informação de certos atributos dos trabalhos, atribuição é feita já seja em uma queue de um centro de trabalho ou na espera para entrar no espaço fabril. Quando um centro de trabalho fica disponível é selecionado o trabalho com maior prioridade para ser processado que se encontrar na queue do mesmo. No caso de a regra de despacho ser aplicada na entrada do espaço fabril, quando é autorizada a entrada, os trabalhos entram para o sistema segundo a sua prioridade. As regras de despacho podem ser classificadas em estáticas ou dinâmicas. As regras estáticas são aquelas em que os valores das prioridades dos trabalhos não variam com o passar do tempo, não são dependentes do tempo. São o resultado de uma função de dados de um trabalho ou posto de trabalho. As regras dinâmicas são dependentes do tempo, nestas regras de despacho a prioridade dos trabalhos muda com

Tabela 2: Lista de regras de despacho

| No. | Rules | Description | Type |
|-----|-------|--------------------------------|---------|
| 1 | FIFO | First In First Out | Static |
| 2 | LIFO | Last In First Out | Static |
| 3 | SPT | Shortest Processing Time | Static |
| 4 | LPT | Longest Processing Time | Static |
| 5 | SPS | Shortest Process Sequence | Static |
| 6 | LPS | Longest Process Sequence | Static |
| 7 | STPT | Shortest Total Processing Time | Static |
| 8 | LTPT | Longest Total Processing Time | Static |
| 9 | ECT | Earliest Creation Time | Dynamic |
| 10 | LCT | Longest Creation Time | Dynamic |
| 11 | SWT | Shortest Waiting Time | Dynamic |
| 12 | LWT | Longest Waiting Time | Dynamic |
| 13 | LTWR | Least Total Work Remaining | Dynamic |
| 14 | MTWR | Most Total Work Remaining | Dynamic |

o passar do tempo. Algumas das mais frequentemente utilizadas regras de despacho estão presentes na tabela seguinte (Kaban et al., 2012).

2.2 SIMULAÇÃO

A distribuição dos centros de trabalho no espaço fabril tem impacto no rendimento geral do sistema de produção. A simulação é uma das ferramentas utilizadas para analisar diferentes alternativas e determinar a opção com melhor performance. Sendo que o desenho do espaço fabril é complexo e específico para cada caso, deu-se o desenvolvimento de diversas metodologias de simulação.

É possível encontrar análises e revisões de métodos de simulação. (Moslemipour et al., 2012), por exemplo, explora e analisa diversas metodologias para o desenho dinâmico e robusto do espaço fabril em sistemas de manufatura flexíveis. Assim como classificações, como (Negahban and Smith, 2014) onde os métodos são classificados segundo a sua utilização e é feita uma revisão das aplicações de simulação em sistemas de produção. Ainda as aplicações da simulação são divididas em análise, desenho e

otimização da gestão industrial, e é ainda enfatizada a importância da simulação nestas áreas.

Na área da gestão da produção, existe um grande número de documentos que apresentam novas metodologias ou abordam a implementação da simulação. (Azadivar and Wang, 2000) apresenta um método que utiliza um algoritmo genético para o desenho do espaço fabril, no sentido de aumentar a eficiência do sistema e simulação de modo a avaliar a performance, a simulação é auxiliada por uma técnica de comparação estatística. O método considera as características dinâmicas e restrições do sistema como um todo. (Jagstam and Klingstam, 2002) recorrem à simulação de eventos discretos para estudar os pré-requisitos de elaboração de uma fábrica virtual para estudar os sistemas de produção. Neste estudo ainda são enfatizados os problemas da utilização de simulação discreta no desenho dos sistemas de produção. (Mendes et al., 2005) utilizam *simulated annealing meta-heuristic* para derivar configurações de linha, posteriormente estas configurações são usadas num modelo de simulação discreta para derivar diferentes medidas de performance para ajustar a configuração de um modelo misto de montagem de câmaras de PC segundo a demanda. (Heilala et al., 2007) introduz um método de simulação de eventos discretos que auxilia no planeamento e na gestão da capacidade da produção, de forma a lidar com as possíveis mudanças no sistema. (Wang et al., 2008) apresentam um método que combina um algoritmo genético e simulação para realizar o esquema do espaço fabril. (Moon et al., 2009) usam simulação para avaliar as mudanças de esquema no espaço fabril de um case study de uma companhia coreana que produz suportes de semicondutores. (Jithavech and Krishnan, 2010) apresentam um método com base em simulação para realizar previsões da incerteza associada com o esquema da linha. É validado o modelo de simulação contra processos analíticos e, posteriormente, são analisados três cenários demonstrando que o método reduz o risco associado com o esquema do espaço fabril. (Azadeh et al., 2011) propõem um algoritmo combinando simulação e *data envelopment analysis* (DEA) capaz de lidar com o problema do esquema do espaço fabril com linhas únicas estocásticas e não lineares com dados ambíguos. O método é demonstrado num *case study* de uma companhia de manufatura de refrigeradores. Em (Michalos et al., 2012), é apresentado um algoritmo de pesquisa inteligente para derivar desenhos da linha de produção, assim como as avaliar face a diversos critérios definidos pelo utilizador. O método é implementado num programa

que automaticamente produz modelos de simulação discreta para analisar o desempenho do sistema. (Ducloux, 2014) usa a simulação para analisar todo o processo de manufatura de engrenagens e prever as propriedades de componentes individuais. (Sardar et al., 2016) recorrem à simulação para encontrar os problemas relacionados com o desenho do espaço fabril e, posteriormente, gerar um desenho ótimo para o sistema de produção da empresa. (Gyulai et al., 2016) apresenta uma metodologia de otimização do desenho do espaço fabril. Os autores utilizam um método automático para a construção de modelos de simulação. Após a avaliação por meio da simulação é utilizado um algoritmo de pesquisa *near-to-optima* para melhorar a solução. (Garcia et al., 2018) aborda a modelagem conceptual da atividade de otimização do desenho do espaço fabril através de simulação em ambientes de grande incerteza. Mostrando como é que critérios de caracterização dos sistemas de produção podem ser usados na modelagem conceptual de forma a reduzir a incerteza. (Zhang et al., 2019) introduz uma metodologia de desenho do espaço fabril à base de simulação e desenvolvem um procedimento de implementação desta metodologia. Estão, ainda, incluídos algoritmos e heurísticas que aplicadas a simulação equilibram a performance da operação e custo do planeamento. Através destes documentos é possível observar que o problema abordado neste documento, assim como a metodologia baseada em simulação, se tem mantido relevante nas últimas duas décadas.

2.2.1 Software de Simulação

A análise de um sistema através de simulação é afetada pela escolha da linguagem de programação ou software de simulação escolhido. Existem diversos softwares de simulação, no entanto, é necessário estar consciente da área para a qual o software foi desenvolvido, assim como se o software se adapta à aplicação pretendida. Podemos considerar dois tipos de software, o software de propósito geral e o software orientado a aplicações. O primeiro pode ser utilizado para qualquer finalidade, podendo incluir ferramentas para aplicar numa área específica. O segundo software é desenvolvido para uma área específica de aplicação como a manufatura ou a saúde. Entre os softwares de propósito gerais mais utilizados temos o ExtendSim, Arena e Simio. Ao considerar a escolha de um software de propósito geral é necessário ter em atenção que se não

existir flexibilidade suficiente ou facilidade de utilização podemos obter resultados incompletos, ou possivelmente erróneos do modelo implementado (Law, 2015).

ExtendSim é o nome que designa o conjunto de quatro software de simulação de propósito geral que possuem as mesmas características básicas e em que cada um está orientado para uma área específica. Neste programa um modelo é criado através da adição de blocos a um worksheet, definir as conexões entre os mesmos e introduzir a informação do sistema para a simulação. Cada bloco tem a sua funcionalidade e características, e cada bloco no modelo tem a sua própria informação. A entidade tem o nome de itens, estas possuem atributos e deslocam-se no modelo. É possível ter uma quantidade quase ilimitada de entidades no modelo devido ao reduzido volume de dados da sua estrutura. Parte da flexibilidade de este software é devida à possibilidade de editar os blocos, sendo que o código destes está disponível para ser editado pelo utilizador, seja de raiz ou modificando os já existentes. Também é possível introduzir qualquer tipo de blocos num bloco hierárquico e guardá-lo na biblioteca para posterior uso (Krahl, 2009; Diamond et al., 2010).

Arena inclui meios de modelagem de processos, análise de dados estatísticos, construção de animações e análise de resultados. É o resultado da combinação da linguagem Siman e do pacote de animação Cinema em 1993. Formado por “modules” e “templates”. Os módulos são o meio de modelagem que contém a lógica mais a interface de utilizador. Os templates como o “Basic Process”, o “Advanced Process”, o “Advanced Transfer” e o “Flow process” são utilizados para organizar os módulos segundo a sua função. Os templates “Blocks” e “Elements” são usados na modelagem de complexos sistemas reais, sendo anteriormente os componentes da linguagem Siman. Os blocos são construções lógicas básicas que representam operações. Os elementos são objetos que representam equipamento como recursos, queues e tallis (Altiok and Melamed, 2007). A versão profissional do Arena inclui a possibilidade de criar módulos personalizados e de os guardar num template. É possível executar o modelo em tempo real, ou um múltiplo do mesmo, e de o manipular por meio de processos.

Simio é uma ferramenta de simulação com base em objetos inteligentes, desenvolvido em 2007. Cada objeto no Simio possui propriedades, states e lógica. É possível modificar objetos e guardá-los para posterior uso (Law, 2015). Ao contrário de outros softwares,

não é necessário utilizar código de programação para criar objetos, sendo um processo gráfico (Sturrock and Pegden, 2010). Não existem diferenças entre um objeto e um modelo, uma vez que as suas criações são idênticas. Um modelo assemelha-se ao sistema real, sendo assim simples de se perceber o seu funcionamento. A lógica e animação do modelo são construídas simultaneamente, tornando o processo de modelagem intuitivo (Pegden, 2007; Pegden and Sturrock, 2011). As animações 3D são incluídas por defeito, basta pressionar uma tecla para trocar de 2D a 3D. A execução iterativa e o modelo experimental são as opções de execução do Simio. A primeira opção é útil para construir e validar o modelo, e é onde se pode assistir às animações de execução do modelo. A segunda opção é utilizada para avaliar a influência da alteração de parâmetros no modelo (Sturrock and Pegden, 2010).

2.3 METAMODELOS

A simulação é um processo dispendioso, isto leva a procura de soluções e alternativas para este problema. Ao longo dos anos foram apresentados diversos métodos e estratégias para otimizar a simulação. Metamodelos, raking and selection, gradient-based procedures, random search, sample path optimization e metaheurísticas, estão incluídos entre estas estratégias (April et al., 2003; Fu et al., 2005). Os metamodelos é um conceito introduzido por Robert Blanning (Blanning, 1974, 1975a,b), com o seu desenvolvimento e popularização realizado por Jack Kleijnen (Kleijnen, 1975). Podemos definir um meta-modelo como uma função, representado por f , que toma alguns parâmetros do modelo de simulação como entradas, representados pelo vetor x , e produz uma aproximação de um conjunto de características de um output da simulação, representado por $g(Y)$. Se o valor de uma característica for $Y(x)$ para uma simulação real executada com os parâmetros definidos com os valores em x , representamos a aproximação do metamodelo $f(x)$, como (Barton, 2015):

$$h(Y(x)) \approx f(x)$$

Um metamodelo não é só um método de estudar o comportamento de um modelo de simulação, sendo também um método de filtrar noisy data (Box and Draper, 1987; Forres-

ter et al., 2008). Um metamodelo tem um conjunto de características que lhe proporciona algumas vantagens sobre a simulação, como o facto de possui uma forma específica, uma resposta específica e a sua computação é pouco dispendiosa(Balaban and Dengiz, 2018). É possível encontrar documentos que estudam diversas técnicas de metamodelagem como regressão polinomial(Kleijnen, 2007; Liu et al., 2016), MARS(multivariate adaptive regression splines)(Li et al., 2010; Chen et al., 2017), SVR (support vector regression)(Wang et al., 2011; Villa-Vialaneix et al., 2012), Neural Network(Can and Heavey, 2012; Kroetz et al., 2017), Radial Basis Function(Fang and Horstemeyer, 2006; Chen et al., 2019), Moving Least Square(Kim et al., 2009; Salemi et al., 2012) e Kriging(Van Gelder et al., 2014a; Østergård et al., 2018). Assim como variações de estes e outros métodos de metamodelagem como stochastic Kriging(Ankenman et al., 2008; Chen and Kim, 2013) e ordinary Kriging(Biles et al., 2007; Zakerifar et al., 2009) para o caso do Kriging ou Bayesian Neural Network(Zhao and Xue, 2010) considerando o método Neural Network. Os metamodelos também foram implementados com diversas finalidades como otimização(Cheng and Currie, 2004; Van Gelder et al., 2014b) estudo das incertezas dos parâmetros do modelo (Xie et al., 2014; Yuan et al., 2017) auxilio na tomada de decisão de desenho de um sistema (Dellino et al., 2009; Hester et al., 2017) e análise de sensibilidade(Zhao and Bu, 2019; Chen et al., 2017).

2.4 ANÁLISE CRÍTICA DA LITERATURA

O problema de escalonamento é complexo o que o levou a ser um constante foco de análise e investigação nas últimas décadas. Foram desenvolvidos diversos métodos para lidar com o problema de escalonamento, sendo a simulação um de estes métodos. A simulação provou ser um método com várias vantagens, assim como limitações. É importante lidar com as limitações da simulação, uma vez que com a introdução de novos paradigmas de produção como a indústria 4.0, onde a complexidade dos sistemas de produção aumenta, enfatiza estas limitações. Para lidar com as limitações da simulação podem ser implementados vários métodos de otimização como os metamodelos que combinam as vantagens da simulação com aspetos dos métodos analíticos. É possível encontrar diversos documentos abordando os metamodelos. Porem, em si, é um campo de estudo que é necessário aprofundar, mais quando consideramos os metamodelos aplicados a produção. Sendo a razão da sua escolha para este trabalho. Da mesma

forma existem poucos documentos onde variáveis como os trabalhos serem libertados para o espaço fabril em *batch* ou que não se assume que os tempos de *set up* não são desprezíveis ou estão incluídos no tempo de processamento são consideradas (Rossi, 2014). Estas condições podem não ser aplicadas em sistemas de produção realistas, existindo uma necessidade de estudar as implicações de este tipo de variáveis no sistema. Tornando-se outra razão do desenvolvimento de este trabalho contribuir para o conhecimento das implicações no sistema destas variáveis.

METODOLOGIA FUNDAMENTADA EM METAMODELOS

Cada um dos diversos métodos de otimização da simulação apresenta vantagens e restrições na sua utilização. Entre os vários métodos a abordagem selecionada é a implementação de metamodelos. A metodologia tenta tirar partido de métodos analíticos para atenuar as limitações da simulação, nomeadamente o tempo e esforço computacional da sua implementação. Pode-se descrever de uma forma simples como o ajuste de uma função aos resultados obtidos da simulação para facilitar análise do modelo. Conseguindo-se assim minimizado o tempo de execução e número de execuções do modelo ao obter aproximações dos resultados num determinado espaço de soluções.

O metamodelo implementado é bem conhecido e testado. No transcorrer da implementação dos metamodelos são abordadas diversos *outputs* do sistema, cada uma com um método de metamodelagem que melhor se adapta ao mesmo. No entanto, no âmbito deste projeto e de forma a permitir as corretas comparações entre os diversos metamodelos é necessário implementar um único método de metamodelagem. Sendo o metamodelo escolhido o que possui documentação que respalda o seu bom despenho em diversos *outputs*.

Neste capítulo é descrito o problema em estudo contendo a exposição das bases do método de metamodelagem, como a sua forma, e definição dos cenários analisados.

3.1 DESCRIÇÃO DO PROBLEMA

O planeamento da produção muitas vezes é associado a previsões temporais de médio e longo prazo. Os objetivos da sua implementação inclui a otimização do desempenho, reduzir custos ou aumentar o *output* de produtos do sistema. É possível lograr estes objetivos com compreensão do comportamento sistema de produção. A simulação destaca-se entre os métodos de estudo de um sistema de produção. Porém, os modelos de simulação são complexos e mesmo para sistemas relativamente pequenos o tempo de execução é elevado. Fundamenta-se assim a implementação de metamodelos neste trabalho.

O problema de escalonamento base é formulado pelos pontos a continuação:

1. Existe um conjunto de n trabalhos que são processados em m máquinas.
2. Os trabalhos são uma sequência de n operações predefinidas.
3. Cada máquina tem um tempo de set up predefinido que depende do produto.
4. Cada operação deve ser realizada para completar o trabalho e cada trabalho tem disponíveis as máquinas para o seu processamento.
5. Todas as máquinas estão disponíveis no tempo de simulação $t = 0$.
6. Os trabalhos são libertados em intervalos predefinidos.
7. Cada item não pode ser processado por mais de uma máquina simultaneamente.
8. Cada máquina só pode realizar uma operação de cada vez.
9. Não existem restrições de antecedência entre as operações de diferentes trabalhos, assim sendo os trabalhos são independentes entre si.
10. Uma operação uma vez iniciada não pode ser interrompida.
11. Os tempos de transporte entre máquinas estão incluídos nos tempos de processamento.

3.2 DEFINIÇÃO DOS EXPERIMENTOS

Após a implementação do modelo é possível analisar cada aspeto do mesmo através de experimentos. Para isto, são definidos os aspetos a analisar que incidem no tema proposto deste documento. O conjunto de inputs selecionados é formado pelas regras de despacho, tempo na pool e tempo de *set up*. Nos experimentos são alterados os valores destas variáveis para obter os distintos cenários.

Entre a infinidade de possibilidades para os valores das variáveis, são selecionados os valores que permitem uma análise coerente dos *inputs* do sistema. Para o tempo na pool são utilizados 5, 10 e 15 minutos. Embora a análise de três valores não permita analisar em pormenor das implicações desta variável, é possível ter um indício de como uma diminuição e aumento no valor se traduz no comportamento do sistema. No caso do tempo de *set up* é considerado que ao alcançar 30% do tempo de processamento o mesmo tem um impacto significativo no sistema. Assim sendo, nos experimentos é feita a variação do tempo de *set up* de 0% a 30% com incrementos de 2%. Resultando em cenários onde é representando o sistema sem a presença de tempo de *set up* e a sua presença com um aumento progressivo. Os experimentos são o conjunto das combinações dos valores anteriores com quatro regras de despacho.

Em adição é criado um conjunto de experimentos para observar o comportamento do sistema e impacto das regras de despacho ao longo do tempo. Neste caso é utilizado o tempo de pool como sendo 10 minutos e a percentagem de *set up* de 15% em união com cada regra de despacho, e sendo alterado o tempo de simulação. A primeira iteração é feita com uma duração de 2 horas, cada iteração aumenta o tempo de execução 2 horas. Finalizando-se a última iteração com um tempo de 24 horas.

3.3 METAMODELO

Metamodelagem, aplicada a simulação, refere-se a substituir um modelo de simulação por uma função que aproxima o comportamento dos seus *outputs*. Ao executar o modelo de simulação é obtido um conjunto relativamente pequeno de *outputs*. O metamodelo surge ao aprender a aproximação do modelo utilizando estes *outputs* e os respetivos

inputs. Um metamodelo permite realizar análises mais complexas ao diminuir o tempo de avaliação do modelo.

Ao existirem diversas técnicas de metamodelagem é necessário considerar qual se adapta ao problema e objetivos abordados. Após o estudo da literatura sobre comparação entre técnicas de metamodelagem, é escolhido a aplicação de kriging tendo como base a metodologia presente em (Kianifar and Campean, 2020) para selecionar a base do metamodelo. Como cada metamodelo apresenta um desempenho diferente dependendo do conjunto de *outputs* abordado é utilizada a forma que apresenta o melhor desempenho para a maior diversidade de *toutputs*.

Na forma mais simples kriging é um método de interpolação estocástico. Funções complexas são interpoladas por meio de processos estocásticos gaussianos. Desenvolvida por Krige (Krige, 1951) como uma ferramenta de interpolação espacial em geoestatística e posteriormente introduzido no contexto de metamodelagem e experimento de computação (Sacks et al., 1989). Posteriormente, surge um aumento no interesse dos processos Gaussiano com a introdução da regressão do processo Gaussiano, a qual suportava dados com a presença de ruído em aplicações de regressão e classificação em *machine learning* (Rasmussen and Williams, 2006).

3.3.1 Kriging

Considerando \mathcal{M} como a representação do comportamento de um sistema que mapeia o espaço de dimensão M dos *inputs* \mathcal{D}_x no espaço de *output* de dimensão 1. Ou da mesma forma, $\mathcal{M} : x \in \mathcal{D}_x \subset \mathbb{R}^M \mapsto y \in \mathbb{R}$ onde $x = \{x_1, \dots, x_M\}^\top$. Um metamodelo *Kriging* é um algoritmo estocástico que assume que a verdadeira resposta do modelo é a realização de um processo estocástico Gaussiano, descrito pela equação (Santner et al., 2003)

$$\mathcal{M}^k(x) = \beta^\top f(x) + \sigma^2 Z(x, w)$$

Na equação, $\beta^\top f(x)$ é o valor médio do processo Gaussiano recebendo o nome *trend*. Consistindo de P funções arbitrárias $\{f_j; j = 1, \dots, P\}$ e os correspondentes coeficientes $\{\beta_j = 1, \dots, P\}$. O termo seguinte compreende a variância do processo Gaussiano σ^2

e um processo estocástico Gaussiano $Z(x, w)$ com média zero e estacionário, isto é a função de covariância de Z depende somente da diferença entre duas variáveis de *input*. O espaço de probabilidade implícito é representado por w , definido em termos de uma função R (família de correlação) e os hiperparâmetros θ . A correlação entre dois pontos da amostra no espaço de *output* é descrita pela função de correlação $R = R(x, x'; \theta)$, que depende dos pontos x, x' e os hiperparâmetros θ .

3.3.2 Predições por Interpolação

Começamos por abordar os metamodelos Kriging sem a presença de ruído nos *outputs* do modelo. O objetivo da metodologia é fazer a predição de $M^k(x)$, sendo x um ponto que não pertence ao conjunto de *inputs*, partindo das observações do experimento $X = \{x^{(1)}, \dots, x^{(N)}\}$ e as correspondentes respostas do modelo, livres de ruído $Y = \{y^{(1)} = M(x^{(1)}), \dots, y^{(N)} = M(x^{(N)})\}$.

Um metamodelo kriging utiliza as propriedades Gaussianas de um determinado processo para obter a predição. A predição num ponto x é dada por uma variável aleatória $\hat{Y}(x) \sim N(\mu_{\hat{Y}}(x), \sigma_{\hat{Y}}(x))$, isto posto a aproximação do modelo computacional é em essência uma família infinita de estes modelos. Na execução a aproximação Kriging toma a forma da resposta media, enquanto a sua variância é usualmente interpretada como a medida do erro local da aproximação.

A suposição Gaussiana afirma que o vetor formado pela predição no ponto x , dada por $\hat{Y}(x)$, e a verdadeira resposta do modelo Y , possuem uma distribuição Gaussiana conjunta definida por:

$$\begin{Bmatrix} \hat{Y}(x) \\ Y \end{Bmatrix} \sim N_{N+1} \left(\begin{Bmatrix} f^\top(x)\beta \\ F\beta \end{Bmatrix}, \sigma^2 \begin{Bmatrix} 1 & r^\top(x) \\ r(x) & R \end{Bmatrix} \right)$$

Sendo F a matriz de informação dos termos genéricos da forma:

$$F_{ij} = f_j(x^{(i)}), i = 1, \dots, N; j = 1, \dots, P.$$

$r(x)$ é o vetor de correlações cruzadas entre a predição no ponto x e cada uma das observações cujos elementos tem a forma

$$r_i = R(x, x^{(i)}; \theta), i = 1, \dots, N.$$

R é a matriz de correlação com dada por:

$$R_{ij} = R(x^{(i)}, x^{(j)}; \theta), i, j = 1, \dots, N.$$

Consequentemente, a média e variância da variável aleatória $\hat{Y}(x)$ sujeitas as observações X e Y pode ser expressa por (Santner et al., 2003):

$$\mu_{\hat{Y}}(x) = f(x)^\top \hat{\beta} + r(x)^\top R^{-1}(Y - F\hat{\beta}),$$

$$\sigma_{\hat{Y}}^2(x) = \sigma^2(1 - r^\top(x)R^{-1}r(x) + u^\top(x)(F^\top R^{-1}F)^{-1}u(x))$$

Estas equações são a média e variância da predição Kriging, onde

$$\hat{\beta} = (F^\top R^{-1}F)^{-1}F^\top R^{-1}Y$$

é a generalização da estimativa dos mínimos quadrados de β e

$$u(x) = F^\top R^{-1}r(x) - f(x)$$

Ao aplicar Kriging a resposta de um modelo sem ruído temos que a variância nos *inputs* $x \in X$ é reduzida a zero. Significando que a predição Kriging toma os valores do modelo nos *inputs* do mesmo.

Ainda, ao considerar o corolário da suposição Gaussiana.

$$\hat{Y}(x) \sim N(\mu_{\hat{Y}}(x), \sigma_{\hat{Y}}^2(x))$$

Temos que,

$$\mathbb{P}(\hat{Y}(x) \leq t) = \Phi\left(\frac{t - \mu_{\hat{Y}}(x)}{\sigma_{\hat{Y}}(x)}\right)$$

Onde $\Phi(\cdot)$ é a função Gaussiana cumulativa de densidade. Sendo a partir de esta realizado o cálculo do intervalo de confiança da predição com as equações.

$$\hat{Y}(x) \in [\mu_{\hat{Y}}(x) - \Phi^{-1}(1 - \frac{\alpha}{2})\sigma_{\hat{Y}}(x), \mu_{\hat{Y}}(x) + \Phi^{-1}(1 - \frac{\alpha}{2})\sigma_{\hat{Y}}(x)]$$

Interpretando-se como o intervalo em que a predição Kriging tem probabilidade $1 - \alpha$.

3.3.3 Predições Através de Regressão

Na hora de lidar com simulação estocástica é importante considerar a presença de ruído nos resultados do modelo. O ruído pode ser definir como uma perturbação aleatória nos dados obtidos. Resultando dos meios de obtenção dos resultados ou das características aleatórias do sistema. Podemos expressar um *output* com presença de ruído na forma seguinte:

$$y = M(x) + \varepsilon$$

Sendo frequentemente suposto que o ruído ε segue uma distribuição Gaussiana com média zero.

$$\varepsilon \sim N(0, \Sigma_n)$$

Em que Σ_n é a matriz de covariância do termo do ruído.

Dependendo das características de Σ_n pode-se classificar o ruído em três classes.

- O ruído é homogêneo ou homoscedástico no caso em que $\Sigma_n = \sigma_n^2 I$, I representando a matriz identidade. Para este tipo de ruído a variância σ_n^2 é igual e constante para todos os *outputs*
- O ruído é heterogêneo independente ou heterocedástico quando $\Sigma_n = \text{diag}(\sigma_n^2)$, a variância do ruído σ_n^2 é diferente para cada *output* e não possuem relação entre si.
- O ruído é heterocedástico geral ao dar-se Σ_n , no qual a variância do ruído σ_n^2 pode divergir para cada *output* e existindo a possibilidade de relação entre *outputs*.

A distribuição Gaussiana ao considerar a predição em x , $\hat{Y}(x)$, e as respostas do modelo Y com presença de ruído toma a forma.

$$\begin{Bmatrix} \hat{Y}(x) \\ Y \end{Bmatrix} \sim N_{N+1} \left(\begin{Bmatrix} f^\top(x)\beta \\ F\beta \end{Bmatrix}, \begin{Bmatrix} \sigma^2 & \sigma^2 r^\top(x) \\ \sigma^2 r(x) & \sigma^2 R + \Sigma_n \end{Bmatrix} \right)$$

Devido a heteroscedasticidade, usualmente a variância do ruído não pode ser fatorizada fora da matriz de covariância, assim a matriz de covariância $C = \sigma^2 R + \Sigma_n$ é utilizada na formulação de média e variância da predição Kriging, resultando nas equações.

$$\mu_{\hat{Y}}(x) = f(x)^\top \hat{\beta} + c(x)^\top C^{-1} (Y - F\hat{\beta}),$$

$$\sigma_{\hat{Y}}^2(x) = (\sigma^2 - c^\top(x)C^{-1}c(x) + u_c^\top(x)(F^\top C^{-1}F)^{-1}u_c(x))$$

Em que $c = \sigma^2 r(x)$ é o vetor de covariâncias cruzadas,

$$\hat{\beta} = (F^\top C^{-1}F)^{-1}F^\top C^{-1}Y$$

É a estimativa generalizada de mínimos quadrados do coeficiente de regressão $\hat{\beta}$, e

$$u_c(x) = F^\top C^{-1}c(x) - f(x)$$

Para o caso do ruído homoscedástico é utilizado $C = \sigma^2 R + \sigma_n^2 I$ em conjunção com as equações.

$$\sigma_{total}^2 = \sigma^2 + \sigma_n^2$$

$$\tau = \frac{\sigma_n^2}{\sigma_{total}^2}$$

Originando a formulação da distribuição Gaussiana conjunta como.

$$\begin{Bmatrix} \hat{Y}(x) \\ Y \end{Bmatrix} \sim N_{N+1} \left(\begin{Bmatrix} f^\top(x)\beta \\ F\beta \end{Bmatrix}, \sigma_{total}^2 \begin{Bmatrix} (1-\tau) & \tilde{r}^\top(x) \\ \tilde{r}(x) & \tilde{R} \end{Bmatrix} \right)$$

Em que,

$$\tilde{r} = (1 - \tau)r$$

$$\tilde{R} = (1 - \tau)R + \tau I$$

Para este caso a média e variância da variável aleatória Gaussiana $\hat{Y}(x)$ é expressa como (Rasmussen and Williams, 2006)

$$\mu_{\hat{Y}}(x) = f(x)^\top \hat{\beta} + \tilde{r}(x)^\top \tilde{R}^{-1}(Y - F\hat{\beta}),$$

$$\sigma_{\hat{Y}}^2(x) = \sigma_{total}^2(1 - \tilde{r}^\top(x)\tilde{R}^{-1}\tilde{r}(x) + u^\top(x)(F^\top \tilde{R}^{-1}F)^{-1}u(x))$$

Onde $\hat{\beta}$ e u seguem as equações definidas na secção anterior com a substituição de R e r por \tilde{R} e \tilde{r} respetivamente.

Em contraste com o caso da ausência de ruído, a variância da predição num *output* do modelo $x \in X$ não é reduzido a zero, convertendo a predição Kriging num modelo de regressão. No entanto, outros corolários da suposição Gaussiana ainda podem ser aplicados.

3.3.4 Trend

Trend é o nome que recebe a média do processo Gaussiano, sendo o termo $\beta^\top f(x)$ da equação apresentada anteriormente. Não é necessário que a *trend* seja diferente de zero, no entanto, na prática é preferível (Rasmussen and Williams, 2006). Em função do tipo de *trend* um metamodelo Kriging recebe um nome diferente.

Um metamodelo Kriging recebe o nome em função do tipo de *trend* utilizada (Dubourg, 2011).

- Simple Kriging, caso a *trend* seja conhecida

$$\beta^\top f(x) = \sum_{j=1}^P f_j(x)$$

Onde os f_j são funções arbitrárias, mas especificadas complementarmente. Ainda nenhuma estimativa de β é realizada, sendo todos os coeficientes 1.

- Ordinary Kriging, a *trend* possui um valor constante porem este é desconhecido.

$$\beta^\top f(x) = \beta_0 f_0(x) = \beta_0$$

Onde por convecção $f_0(x) = 1$.

- Universal Kriging, sendo a formulação mais geral e flexível é assumido que a *trend* resulta da soma de P funções arbitrárias $f_k(x)$,

$$\beta^\top f(x) = \sum_{k=1}^P \beta_k f_k(x)$$

Ao analisar as equações anteriores, é possível concluir que o simple e ordinary Kriging são casos especiais do universal Kriging.

3.3.5 Função de Correlação

A função de correlação é o componente que contem as suposições sobre a função de aproximação. Descreve a semelhança entre os *outputs* do modelo e um novo ponto, dependendo da distância dos *inputs*. É assim, esperado obter valores similares para *inputs* próximos.

Uma função arbitrária (x, x') não é, geralmente, uma função de correlação valida. Para ser valida uma função de correlação $R(x, x')$ tem que satisfazer certas condições. Uma unção de correlação valida é simétrica, $R(x, x') = R(x', x), \forall x', x \in D_X$ e a matriz de correlação com elementos $R_{ij} = R(x^{(i)}, x^{(j)}), (x_i, x_j) \in X \times X$ é positiva semi-definida para qualquer conjunto de pontos N dos resultados X , $N^\top M N \geq 0, \forall N \in X$ com M sendo a matriz de correlação, simétrica e real.

Comummente expressa-se a função de correlação na forma $R(x, x'; \theta)$ sendo θ um vetor contendo um conjunto de parâmetros. Usualmente $\theta \in \mathbb{R}^M$, o que pode não ser verdade no caso geral, uma vez que mais de um parâmetro pode corresponder a cada

dimensão de *input*. Porém em secções seguintes é assumido que um elemento de θ é utilizado por dimensão de *input*.

Sendo que os hyperparametros θ são desconhecidos é necessário fazer a sua aproximação de forma a obter o metamodelo Kriging. A aproximação é alcançada ao resolver um problema de otimização, o qual difere para cada método e com o facto de se os *outputs* contenham ruído ou não. Os detalhes do metamodelo implementado são apresentados no capítulo subsequente.

DESENVOLVIMENTO DO MODELO

Um dos pontos fulcrais do modelo é a família de produtos. Esta designação refere-se a um conjunto de produtos em que a sua sequência é composta pelos mesmos passos, divergindo na ordem ou tempos de processamento. A primeira sequência criada foi a da entidade *E1*, onde simplesmente foi decidido um número de centros de trabalho com nomes arbitrários que componham a sequência desta entidade. De forma a obter o resto das entidades e alterando a sequência da entidade *E1* ao repetir ou remover centros de trabalho. O resultado é uma família de produtos com uma notável heterogeneidade.

Tabela 3: Sequência de processamento dos produtos

| Entidade | Sequência |
|----------|----------------------|
| E1 | CC-DC-AC-JC-PC |
| E2 | CC-JC-DC-AC-PC-JC-PC |
| E3 | CC-DC-CC-AC-JC-PC |
| E4 | CC-DC-AC-JC-AC-PC-DC |
| E5 | CC-DC-JC-PC |

O modelo está implementado em Simio, onde a base do modelo neste software são objetos. Começa-se assim, por criar a distribuição de objetos que representam o problema descrito no capítulo anterior. Portanto, a partir das sequências são obtidas as duas primeiras configurações de linha ao aplicar uma heurística simples, sendo o processo utilizado devido à existência da restrição de não saltar máquinas. Onde o primeiro passo passa por organizar os trabalhos pelo comprimento da sequência. Coloca-se as workstations segundo a sequência da primeira entidade na lista e de seguida verifica-se por ordem da sequência da entidade seguinte se a workstation esta disponível, verificando a existência de uma workstation a começar pela workstation onde é realizado o processamento do passo anterior da sequência. Caso a workstation exista,

passamos para o seguinte passo da sequência caso contrário inserimos a workstation na linha. Na configuração com reentrada, as entidades podem entrar na linha de produção em qualquer ponto a configuração de linha toma a forma da sequência da entidade E1. Não existe a necessidade de ter mais de uma workstation de cada tipo. Algo similar acontece com a linha paralela, no entanto, nesta configuração de linha é identificada a posição do *bottleneck*. Neste ponto são colocadas duas workstations que processam em paralelo, realizam o mesmo processo simultaneamente.

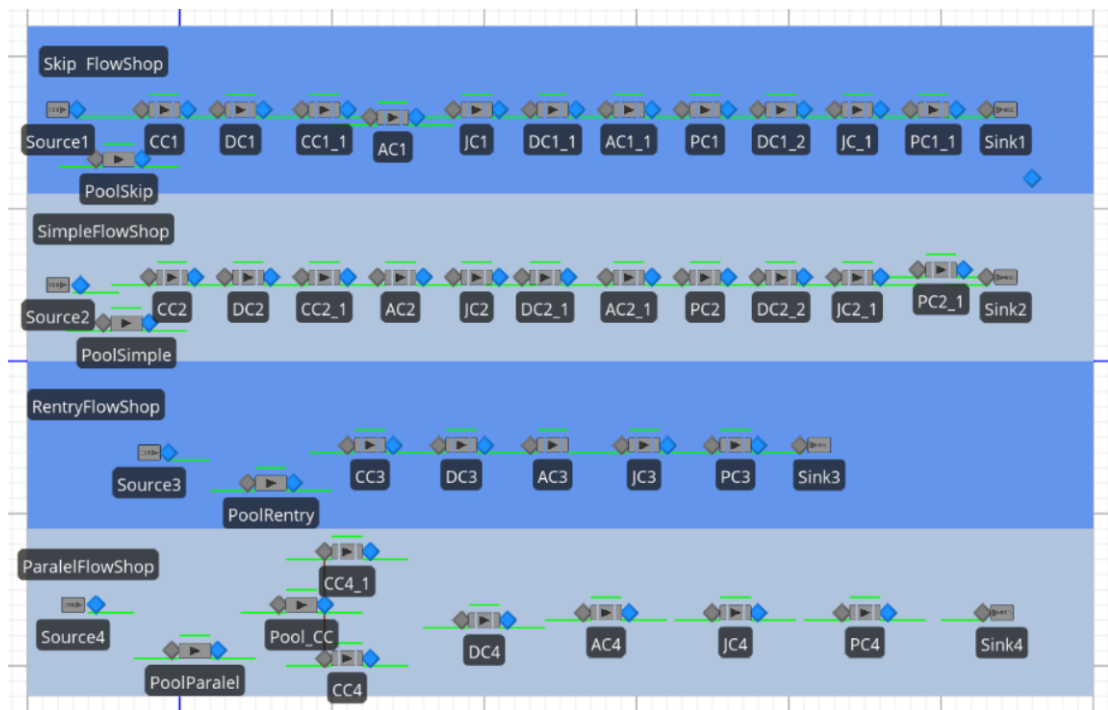


Figura 1: Distribuição dos centros de trabalho

4.1 DEFINITIONS

No Simio as *definitions* são um conjunto de variáveis com características e comportamentos predefinidos, utilizadas para guardar algum tipo de informação utilizada pelo modelo ou experimentos. No modelo foram utilizados três tipos de variáveis distintas *states*, *propertys* e *elements*. Uma característica de todas estas variáveis é que é possível definir o tipo de unidade utilizada, nem todas as variáveis definidas tenham o tipo de unidade definida. As características e conjunto de variáveis definidas de cada tipo são

apresentadas, sendo posteriormente abordadas individualmente no contexto em que são aplicadas e utilizadas.

4.1.1 States

Um *state* é uma variável que pode mudar de valor durante a simulação. Podem ser discretos, quando o seu valor é alterado ao se atribuir uma lógica em certos momentos durante a simulação, ou contínuos, o seu valor é alterado continuamente e automaticamente seguindo alguns dos seus atributos. Todos os *states* definidos são discretos. Dependendo da secção do projeto em que foram implementados temos dois conjuntos de *states*. O conjunto de *states* definidos na secção do modelo.

| State Variables | | |
|-----------------|------------------------|--------------|
| IndexSPT | Integer State Variable | IndexSPT |
| N_atrasos | Real State Variable | N_atrasos |
| N_adiantados | Real State Variable | N_adiantados |
| Interval | Real State Variable | Interval |
| IndexLTPT | Integer State Variable | IndexLTPT |
| IndexSPS | Integer State Variable | IndexSPS |
| IndexEDD | Integer State Variable | IndexEDD |
| LastEnt1 | Integer State Variable | LastEnt1 |
| LastEnt2 | Integer State Variable | LastEnt2 |
| LastEnt3 | Integer State Variable | LastEnt3 |
| LastEnt4 | Integer State Variable | LastEnt4 |

Figura 2: States definidos no Model

A secção ModelEntity, é onde estão contidas todas as propriedades, *states* e comportamentos que uma entidade possui. Sendo onde foram definidos os *states* que são atribuídos as entidades durante a simulação.

| State Variables | | |
|---------------------|--------------------------|---------------------|
| Picture | Real State Variable | Picture |
| Animation | String State Variable | Animation |
| DueDate | Date Time State Variable | DueDate |
| TempoEntradaSistema | Real State Variable | TempoEntradaSistema |
| IndexSetUp | Integer State Variable | IndexSetUp |

Figura 3: States definidos no ModelEntity

Um aspeto importante dos *states* é que a sua dimensão pode ser alterada. Sendo utilizados *states* como escalares

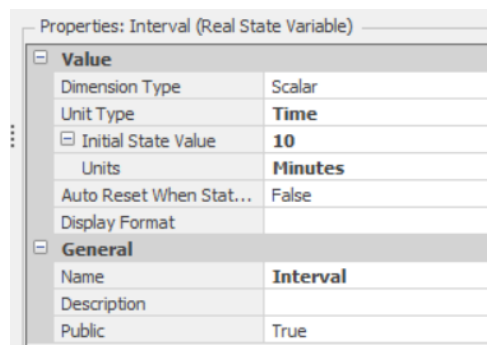


Figura 4: Exemplo de state definido como escalar

e vetores.

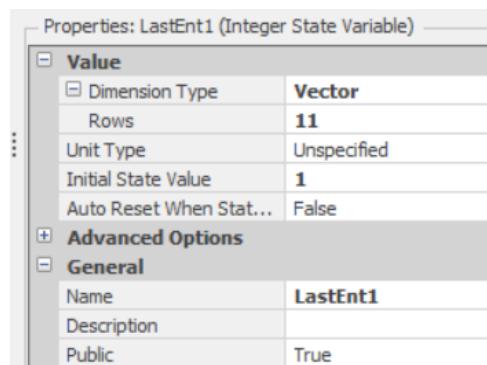


Figura 5: Exemplo de state definido como vetor

4.1.2 Properties

As *properties* são parâmetros de input estáticos para um objeto, o seu valor não é alterado durante a simulação. Podem ser implementados diversos tipos de *properties*, porem todas as *properties* implementadas são *Standard Properties*. Significando que guardam um tipo específico de dados, números inteiros e reais no caso do modelo implementado.



Figura 6: Conjunto de propriedades

Os diversos cenários dos experimentos obtêm-se ao utilizar estas três *properties* em conjunto com modificações no tempo de simulação. O que quer dizer que os valores das *properties* e tempo de simulação caraterizam cada cenário dos experimentos.

4.1.3 Elements

Um elemento é um componente do modelo com propriedades, *states* e comportamentos predefinidos. A sua função no modelo é guardar o valor de algumas mediadas de desempenho. Todos os *elements* definidos são incluídos automaticamente nos resultados dos experimentos.

| Output Statistic Elements | |
|---------------------------|--------------------------|
| PInSchedule_1 | Output Statistic Element |
| PInSchedule_2 | Output Statistic Element |
| PInSchedule_3 | Output Statistic Element |
| PInSchedule_4 | Output Statistic Element |
| Tally Statistic Elements | |
| FlowTime1 | Tally Statistic Element |
| FlowTime2 | Tally Statistic Element |
| FlowTime3 | Tally Statistic Element |
| FlowTime4 | Tally Statistic Element |
| Lateness_1 | Tally Statistic Element |
| Lateness_2 | Tally Statistic Element |
| Lateness_3 | Tally Statistic Element |
| Lateness_4 | Tally Statistic Element |
| Tardiness_1 | Tally Statistic Element |
| Tardiness_2 | Tally Statistic Element |
| Tardiness_3 | Tally Statistic Element |
| Tardiness_4 | Tally Statistic Element |

Figura 7: Conjunto de elementos

Temos os *elements* são separados segundo a sua utilização. Output Statistic Elements são utilizados para avaliar e exibir o valor de uma expressão no final da simulação. Neste caso faz o cálculo da percentagem de entidades finalizadas sem atraso, ao dividir o número de trabalhos concluídos na *due date* pelo número total de trabalhos e multiplicando este valor por 100. Tally Statistic Elements são utilizados pelos processos que fazem cálculo das estatísticas do modelo, sendo abordados na secção 4.3.

4.2 OBJETOS

Na construção do modelo foram abordados cinco tipos de objetos. Entidades representam peças, partes, operadores ou produtos. O Source é o ponto de origem das entidades. Workstation toma o lugar das estações de trabalho onde as entidades são processadas. Os servers são utilizados como pools, onde as entidades são reunidas e libertadas em intervalos para o espaço fabril. Sink é o ponto de saída do espaço fabril para as entidades. As especificações de cada objeto do modelo são apresentadas nesta secção. Um ponto em comum, de todos os objetos, é a capacidade ilimitada de todos os buffers, tanto de entrada como saída. Assim enquanto um objeto está retido as entidades que chegam ao mesmo são colocadas em espera até a atividade que está a decorrer finalizar.

4.2.1 Entidades

Estão implementadas cinco entidades distintas associadas as várias tabelas que contribuem para a sua caracterização.

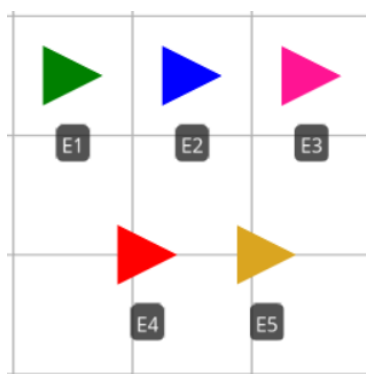


Figura 8: Entidades

Uma desta tabela sendo a tabela Entidades, que se divide nas colunas:

- Entidades, sendo composta pelos nomes das entidades. Ainda tem a função de ser uma chave que referencia as entidades em outras tabelas.
- Percentagem, considerando o total de entidades produzidas representa a percentagem de cada tipo de entidade que é criado, sendo a sua soma é 100.

- Fator Processamento, modificador dos tempos de processamento de cada entidade. Os tempos de processamento de cada entidade é o resultado de multiplicar o valor base do centro de trabalho por este valor.
- NOperações, número de operações necessárias para finalizar o processamento da entidade.
- Media Tempo Processamento Total, media do tempo necessário para finalizar o processamento da entidade. Para o cálculo são utilizadas as médias dos tempos de processamento dos diversos centros de trabalho, sem incluir o Fator Processamento.

| Entidades | Servers | Indicador | Sequence Table_E1 | Sequence Table_E2 | Sequence Table_E3 | Sequence Table_E4 |
|-----------|----------|-------------|---------------------|-------------------|---------------------------------|-------------------|
| | Entidade | Percentagem | Fator Processamento | NOperacoes | Media Tempo Processamento Total | |
| 1 | E1 | 26 | 1 | 5 | 299 | |
| 2 | E2 | 12 | 0.8 | 7 | 409 | |
| 3 | E3 | 15 | 0.7 | 6 | 390,5 | |
| 4 | E4 | 27 | 0.9 | 7 | 396,5 | |
| 5 | E5 | 20 | 1.4 | 4 | 246,5 | |
| * | | | | | | |

Figura 9: Tabela entidades do modelo

Cada Entidade possui uma tabela que define a sua sequência para cada configuração de linha. A sequência é criada ao introduzir os inputs das workstations na ordem correta.

| | Sistema | Entidades | Servers | Indicador | Sequence Table_E1 | Sequence Table_E2 |
|----|-------------|-----------|----------|-----------|-------------------|-------------------|
| | Sequence | Sistema | Entidade | Indicador | | |
| 1 | Input@CC1 | 1 | E1 | 1 | | |
| 2 | Input@DC1 | 1 | E1 | 1 | | |
| 3 | Input@AC1 | 1 | E1 | 1 | | |
| 4 | Input@JC1 | 1 | E1 | 1 | | |
| 5 | Input@PC1 | 1 | E1 | 1 | | |
| 6 | Input@Sink1 | 1 | E1 | 1 | | |
| 7 | Input@CC2 | 2 | E1 | 1 | | |
| 8 | Input@DC2 | 2 | E1 | 1 | | |
| 9 | Input@CC2_1 | 2 | E1 | 0 | | |
| 10 | Input@AC2 | 2 | E1 | 1 | | |
| 11 | Input@JC2 | 2 | E1 | 1 | | |
| 12 | Input@DC2_1 | 2 | E1 | 0 | | |
| 13 | Input@AC2_1 | 2 | E1 | 0 | | |
| 14 | Input@PC2 | 2 | E1 | 1 | | |

Figura 10: Tabela de sequências da entidade 1

Podem ser observadas três *Foreign Keys* na tabela. Sistema, faz referência a configuração de linha e serve para facilitar a visualização da sequência. Entidades, indica a entidade a qual a sequência pertence. Indicador faz referência a tabela com o mesmo nome.



| Sistema | Entidades | Servers | Indicador |
|---------|---|---------|-----------|
| |  Indicador | | |
| 1 |  | 1 | |
| 2 |  | 0 | |
| * | | | |

Figura 11: Tabela Indicador

Os valores da tabela são utilizados na sequência para indicar se entidade é processada na workstation, ao serem incluídos no cálculo do tempo de processamento. Se a entidade é processada o indicador toma o valor de 1 e o caso contrário. A necessidade de este passo deriva-se da restrição de algumas linhas de não haver a possibilidade de saltar centros de trabalho, portanto, a entidade passa por todas as workstations da linha, mesmo nas workstations onde não é processada.

4.2.2 Source e Sink

Cada linha tem como ponto de partida um source e como ponto final um sink. Independentemente da configuração de linha, os source estão configurados de forma idêntica. No entanto, no início da simulação três processos são executados por um source cada um, em que é uma escolha arbitrária definir esta ligação. Dado que não estão ligados ao funcionamento dos source podendo a execução dos processos ser desencadeada por qualquer objeto do modelo. O conjunto de entidades e tipo de entidade criada é definido com recurso a tabela Entidades. Onde é criada uma entidade da tabela Entidades tomando a coluna percentagem como pesos é selecionada uma linha aleatória da tabela. O tempo entre a criação de entidade é um valor aleatório seguindo uma distribuição exponencial com média 80.

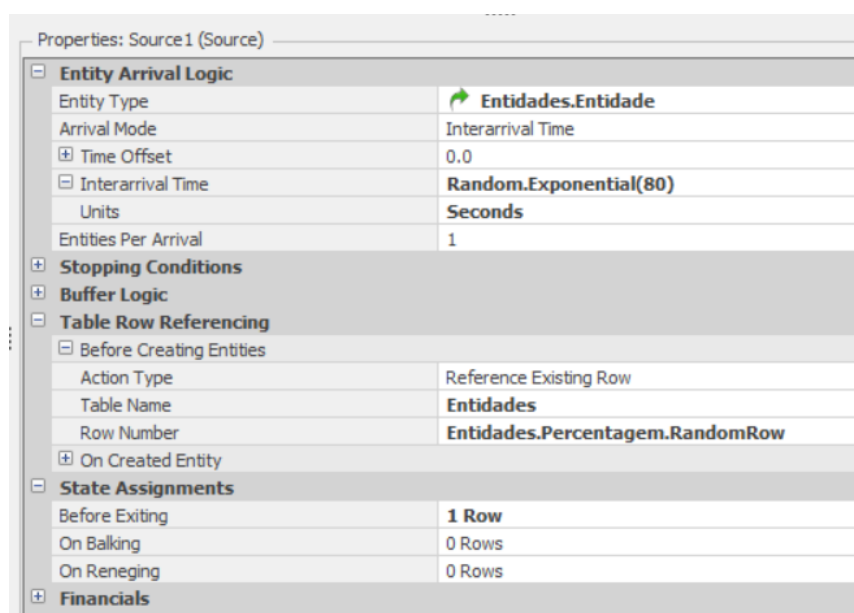


Figura 12: Exemplo da definição de um source

Antes de uma entidade sair de um source é atribuída a *due date*.

O *state Due Date* é um vetor com quatro entradas que aceita valores de tempo e data. Seguindo uma distribuição uniforme é obtido um valor entre 1500 e 3500, representando tempo gasto por operação. Assim este valor é multiplicado pelo número de operações da entidade, obtendo-se a estimativa do tempo máximo utilizado para

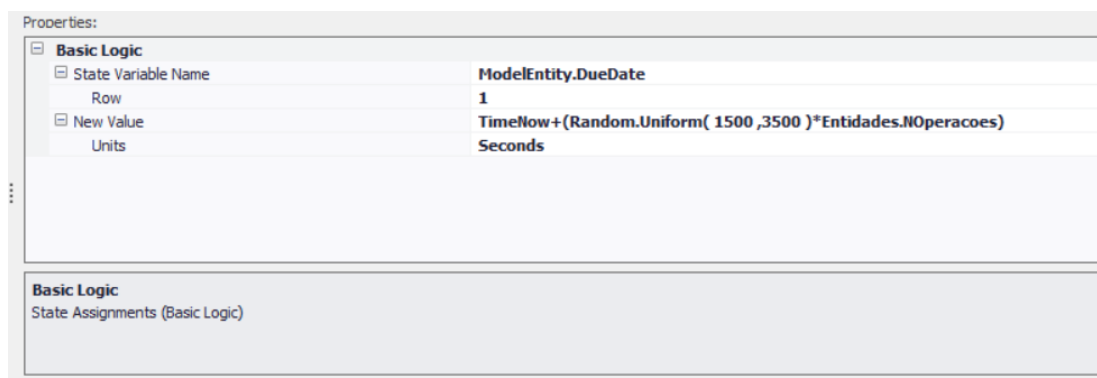


Figura 13: Atribuição da data de entrega

finalizar o processamento. Para finalizar a estimativa, sendo um valor em segundos, é adicionada ao tempo de simulação para obter a *due date*.

As entidades ao sair de um source passam pelo nodo de saída do mesmo. Aqui recebem a prioridade e índice, que posteriormente é utilizado para determinar o tempo de *set up*.

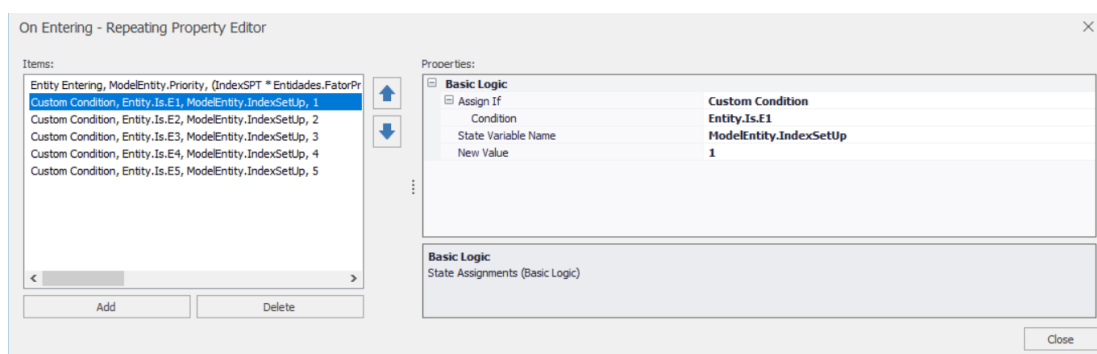


Figura 14: Atribuição da prioridade e index de set up

No fim de cada configuração de linha existe um sink onde as entidades são destruídas. Uma entidade ao entrar no sink ativa-se um processo que realiza o cálculo de algumas medidas de desempenho. O *throughput* dos sinks é utilizado como uma medida de desempenho.

4.2.3 Pool

A pool de cada linha é representada por um server, onde através das suas definições e efeitos de um conjunto de processos, as entidades são reunidas e libertadas para processamento em intervalos. As entidades são ordenadas segundo o seu *state* priority enquanto esperam no nodo de input destes servers, sendo a entidade com menor valor de priority a entidade que possui maior prioridade.

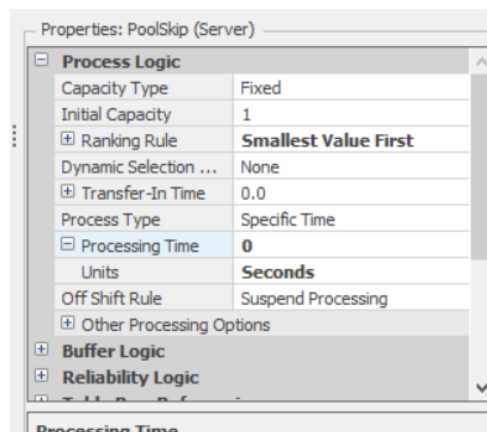


Figura 15: Definições de seleção da prioridade

Para não existirem atrasos na entrada das entidades no sistema e uma vez que não é necessário separar as entidades, o tempo de processamento das pools é de 0 segundos. Significando que chegado o momento todas as entidades no input das pools são libertadas simultaneamente, passando pelos nodos de output fazendo-se a atribuição do *state* TempoEntradaSistema.

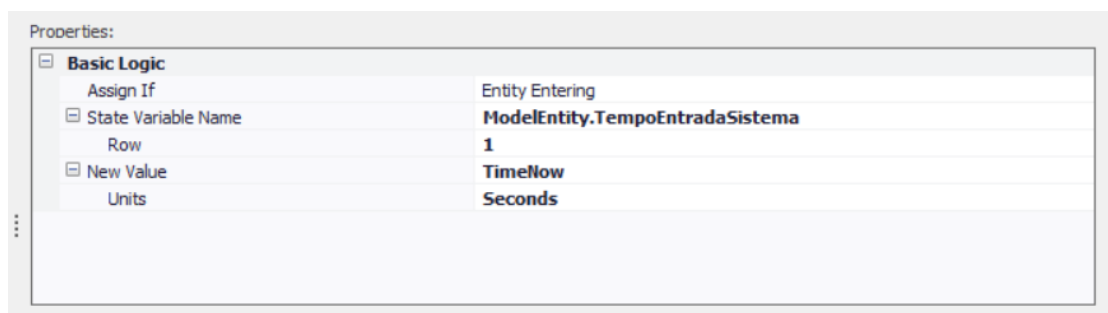


Figura 16: Tempo de entrada no sistema

É também ao entrar neste *outputs* que se executa os processos que seleciona a tabela e linha onde a sequência da entidade dá início. Por exemplo, para a entidade E2 a sua sequência faz parte da tabela SequenceTable_E2. Se a entidade se encontrar na primeira configuração de linha a sua sequência inicia na linha 0 da tabela, por outro lado, se considerarmos a segunda configuração de linha a sequência de E2 começa na linha 8. Assim as entidades ao deixarem estes outputs podem seguir a sequência, estando definido no output.

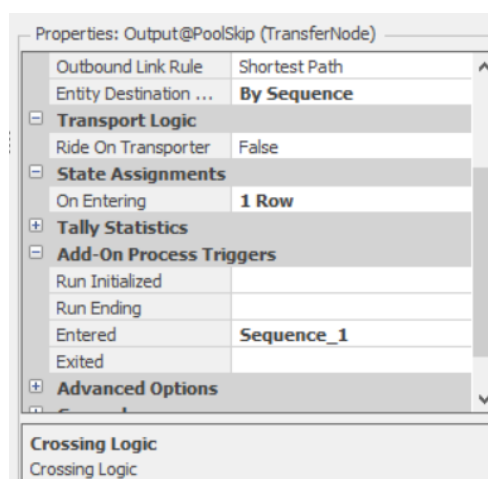


Figura 17: Sequenciamento

Na quarta linha, linha paralela, existem duas pools. Uma definida da mesma forma que as anteriores fazendo o controlo de entidades no espaço fabril. E outra que faz a repartição de entidades entre as Workstations CC4 e CC4.1, máquinas paralelas onde é realizado o mesmo tipo de processamento em simultâneo.

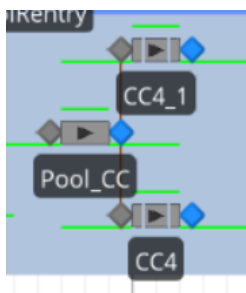


Figura 18: Centros de trabalho paralelos

As entidades são divididas dependendo do número de entidades no input das workstations. A pool tem a configuração por defeito de um server no Simio, alterando-

se unicamente o tempo de processamento para 1 segundo. Ao aplicar este tempo de processamento é possível separar as entidades e que as mesmas cheguem ao input da workstation, evita-se assim problema com a execução do processo que controla a separação de entidades enquanto uma entidade ainda esta a viajar para um input de uma das workstations ou executar o processo para um conjunto de entidades simultaneamente.

Note-se que quando as pools são abordadas como um conjunto a Pool_CC não é incluída.

4.2.4 Workstation

Utilizado para representar os centros de trabalho, O objeto Workstation inclui a possibilidade de uma fácil implementação do tempo de *set up*. No total foram implementadas 33 workstations no modelo, sendo divididas em cinco tipos CC, DC, AC, JC e PC. As entidades são processadas por ordem de chegada, seguem assim a regra FIFO, o que se deve a que a implementação das regras de despacho é realizada nas pools. Sem tomar em conta os processos relacionados com cada workstation, as diferenças entre as configurações é a definição de tempo de *set up* e distribuição uniforme utilizada no cálculo dos tempos de processamento.

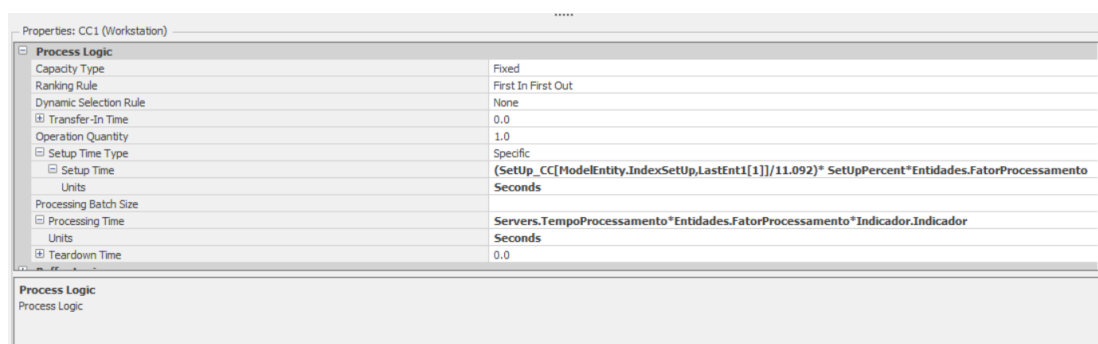


Figura 19: Definição de um centros de trabalho

Para a implementação do set up existe uma tabela para cada tipo de Workstation. Os vários valores aleatórios foram obtidos através de aplicação para gerar números aleatórios, onde foram definidos os limites 5 e 15.

A seleção do valor da tabela empregue para o cálculo do tempo de *set up* é feito através do *state* indexSetUp da entidade selecionada para processamento e valor do

| Set Up_CC | Sequence Table_E2 | | | Sequence Table_E3 | | |
|-----------|-------------------|----|----|-------------------|----|--|
| | E1 | E2 | E3 | E4 | E5 | |
| 1 | 0 | 11 | 15 | 6 | 8 | |
| 2 | 7 | 0 | 8 | 7 | 11 | |
| 3 | 15 | 9 | 0 | 11 | 5 | |
| 4 | 14 | 15 | 13 | 0 | 12 | |
| ► 5 | 8 | 7 | 6 | 15 | 0 | |
| * | | | | | | |

Figura 20: Exemplo de tabela Set Up

mesmo *state* da última entidade processada. Uma das características do modelo é o controlo do set up em percentagens em função do tempo de processamento. Assim, é necessário ajustar o valor proveniente da tabela para que em conjunto com a *property* *SetUpPercent* se obtenha qualquer percentagem desejada. Para isto é calculado o número que ao dividir a média de tempos de set up de uma tabela resulta em 1% da média de tempo de processamento. Ao valor final do tempo de *set up* em segundos obtém-se a fazer o cálculo de:

$$\frac{\text{Valor tabela}}{\text{Valor conversão}} \times \text{SetUpPorcente} \times \text{FatorProcessamento}$$

O tempo de set up não é uma percentagem exata do tempo de processamento, mas é próximo da percentagem pretendida. Sendo que a intenção não é obter valores exatos, mas que o valor difira dependendo das entidades e seja próximo de uma percentagem do tempo de processamento. Tendo-se assim tempos de set up diferentes para as combinações das entidades.

A tabela *Servers* contem todas as *Workstations* do modelo assim como a expressão pela qual se obtém o seu tempo de processamento.

Deste modo o tempo de processamento, em segundos, resulta de um valor aleatório contido nos limites da distribuição uniforme, *FatorProcessamento* da entidade e indicador da sequência.

| Servers | Indicador | Sequence Table_E1 | Set Up_CC | Sequence Table_E2 | Sequ |
|---------|-----------|------------------------|-----------|-------------------|------|
| | Server | Tempo Processamento | | | |
| 1 | CC1 | Random.Uniform(72,111) | | | |
| 2 | CC1_1 | Random.Uniform(72,111) | | | |
| 3 | DC1 | Random.Uniform(30,60) | | | |
| 4 | DC1_1 | Random.Uniform(30,60) | | | |
| 5 | DC1_2 | Random.Uniform(30,60) | | | |
| 6 | AC1 | Random.Uniform(40,65) | | | |
| 7 | AC1_1 | Random.Uniform(40,65) | | | |
| 8 | JC1 | Random.Uniform(40,70) | | | |
| 9 | JC_1 | Random.Uniform(40,70) | | | |
| 10 | PC1 | Random.Uniform(30,80) | | | |
| 11 | PC1_1 | Random.Uniform(30,80) | | | |
| 12 | CC2 | Random.Uniform(72,111) | | | |
| 13 | CC2_1 | Random.Uniform(72,111) | | | |

Figura 21: Tabela de expressões dos tempos de processamento

Finalizado o processamento em qualquer workstation é executado o processo que guarda o IndexSetUp da entidade e por definição no nodo de saída a entidade segue para o próximo passo da sua sequência.

4.3 PROCESSOS

É simples para o utilizador implementar um modelo em Simio, o modelo está pronto a ser executado após criar a distribuição de objetos e definir as propriedades dos mesmos. Esta é a base do modelo onde através de processos são implementadas regras e comportamentos que modificam o modelo durante a sua execução. Um processo é um conjunto de lógicas que são vinculadas aos objetos do modelo, onde na ocorrência de um determinado evento o processo é executado alterando comportamentos ou propriedades do modelo. Estando estruturados como fluxogramas não é necessária a utilização de uma linguagem de programação para a sua implementação. *Steps*, *elements* e *tokens* são os componentes dos processos. Os *tokens* fluem através de um processo executando *steps* que alteram o estado de um ou mais elementos.

A combinação dos objetos e processos possibilita a implementação de sistemas complexos em Simio. Cada objeto tem diversos *triggers* disponíveis, sendo possível

para um objeto utilizar mais de que um processo. Por sua vez um processo pode ser utilizado por vários objetos.

4.3.1 Controle Pool

Por meio de um conjunto de processos é controlada a retenção e liberação de entidades nas pools. Um dos requisitos do problema é que o tempo na pool possa ser modificado nos experimentos. A *property* HoldTime conjugada com o processo TimeDef fazem a especificação do tempo na pool.

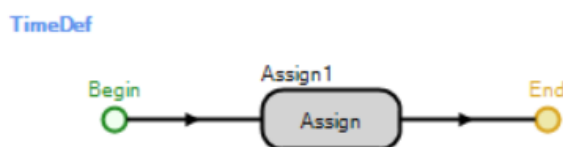


Figura 22: Processo TimeDef

Ao se tratar de uma *property*, HoldTime, pode desempenhar o papel de controle nos experimentos. E através do processo o valor do HoldTime é atribuído ao *state* Interval. O processo é executado uma única vez no início da simulação, estando ligado ao source2.

O *state* Interval é o tempo utilizado pelo processo PoolTime, sendo o processo que controla a liberação das entidades.

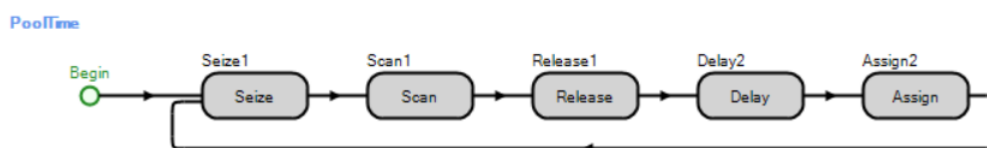


Figura 23: Controle do tempo na pool

O processo divide-se nos seguintes passos ou *steps*

- Seize - Retém a capacidade das quatro pools, não sendo possível para as mesmas processar entidades.

- Scan - Avalia, constantemente, se o tempo de simulação é maior ou igual ao *state Interval*. Enquanto esta condição não for satisfeita o resto do processo não é executado.
- Release - Disponibiliza a capacidade das pools. Uma vez que o tempo de processamento é 0, todas as entidades nos inputs das pools são libertados simultaneamente para o espaço fabril.
- Delay - Atrasa a execução do resto do processo em 0.00000001 segundos. Sem a existência de este passo as entidades não seriam libertadas devido que após a libertação das capacidades das pools as mesmas seriam imediatamente retidas. Devendo-se que este processo é um *loop* executado durante toda a simulação.
- Assign - O valor do *state Interval* é alterado. Ao prévio valor de Interval e adicionado o valor do HoldTime.

Ligado ao Source1 o processo é ativado no início da simulação.

Pool CC

Para o caso da ocorrência especial de uma pool na configuração de linha paralela é necessário um processo que controle o seu comportamento. Como já referido, embora seja uma Pool a mesma é distinta das restantes sendo que somente realiza a distribuição de entidades entra as workstations do tipo CC da configuração de linha. O processo executa-se uma vez que uma entidade entra no output da Pool.CC.

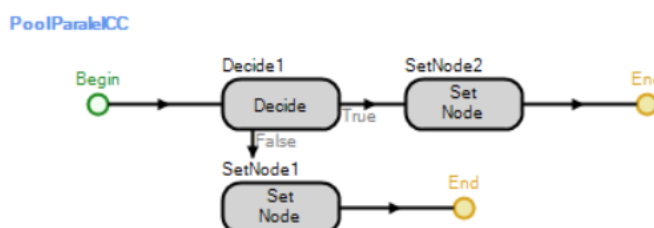


Figura 24: Controle do destino dos produtos na pool das maquinas paralelas

- Decide - Verifica se o número de entidades no input buffer CC4 é maior que o número de entidades no input buffer CC4_1.

- SetNode2 - Caso a condição do Decide se verifique a entidade é enviada para a workstation CC4_1.
- SetNode1- Caso contrário a entidade dirige-se a workstation CC4.

4.3.2 Seleção da Regra de Despacho

As regras de despacho são aplicadas nos inputs buffer das Pools. Entre quatro opções possíveis o processo RuleSelection seleciona a regra a ser aplicado a partir do valor da *property* DispatchingRule. O processo determina qual *state* índice das regras de despacho recebe o valor 1, os restantes recebendo o valor 0. Resultando em que partes específicas da expressão de prioridade sejam anuladas, assim sendo ainda que a expressão de prioridade combina todas as regras de despacho numa fórmula unicamente uma regra de despacho é aplicada. Posteriormente as regras de despacho e a sua formulação são abordadas independentemente com mais detalhe. Ligado ao *trigger* do source 3, o processo é executado uma vez no início da simulação.

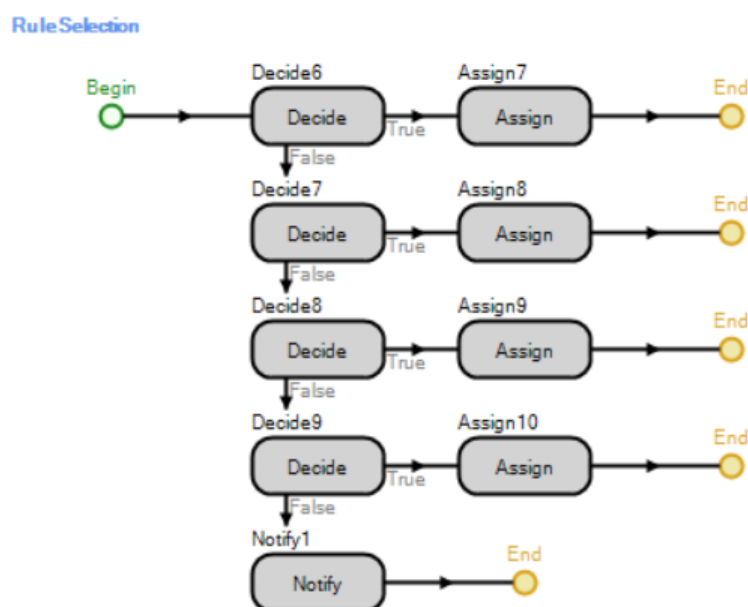


Figura 25: Seleção da regra de despacho

- Decide6 - Verifica se o valor da *property* DispatchingRule é 1. Caso a condição não se cumpra é executado o decide seguinte.

- Assign7 – Caso a condição do decide6 seja cumprida o indexEDD recebe o valor 1. É atribuído o valor 0 aos indexSPT, IndexSPS e IndexTPT.
- Notify - Um step opcional que apresenta uma mensagem de erro se o valor da DispatchingRule não for 1,2,3 ou 4.

Os restantes ramos do processo possuem uma lógica análoga à exposta com o Decide6 e Assign7, diferindo no *state* ao que atribuem o valor 1. Para os Assigns 8,9,10 é atribuído o valor 1 ao IndexTPT, IndexSPS e IndexSPT respetivamente. Sendo que os decides verificam o valor da DispatchingRule na ordem ascendente do 1 ao 4.

4.3.3 Seleção da Sequencia

Ao entrar no *output* de uma pool as entidades executam o processo que faz a seleção da linha, da tabela de sequência correspondente a entidade, onde a sequência de processamento da entidade inicia.

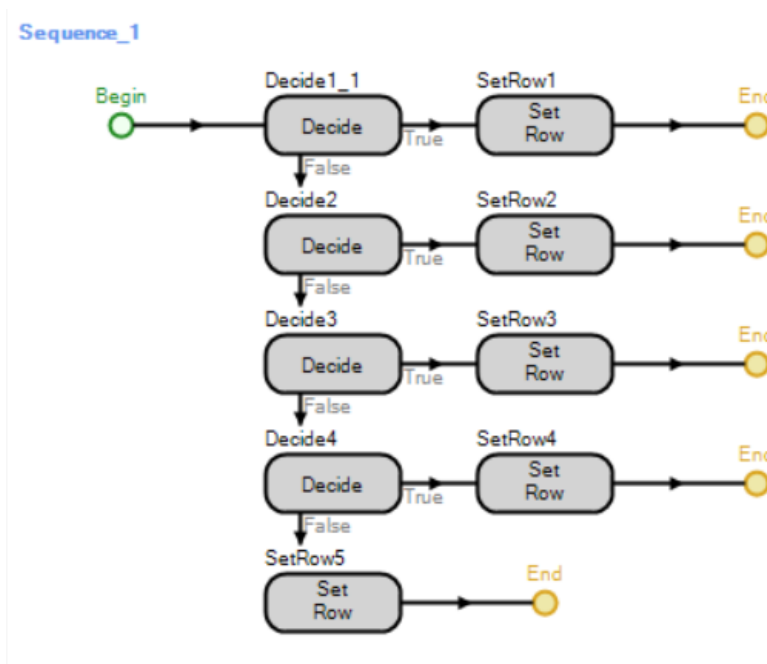


Figura 26: Processo de seleção da sequência

Por ordem ascendente os decides verificam o tipo da entidade que executou o processo, começando na entidade E1 e terminando em E4. Quando a condição de um

decide é satisfeita o Set Row relacionado indica a tabela e linha da mesma onde a sequência de processamento começa. Por exemplo, no processo Sequence_1 e para uma entidade do tipo E1 o processo define o início da sua sequência como sendo a linha o da tabela SequenceTable_E1. Para o mesmo tipo de entidade no processo Sequence_2, implementado na segunda configuração de linha, a sua sequência tem começo na linha 6 da mesma tabela.

4.3.4 Set Up

Um dos passos necessários na seleção do tempo de *set up* é guardar o tipo da última entidade processada nas várias workstations. Para isto ao se finalizar o processamento e antes de a entidade deixar a workstation o processo que guarda o tipo de entidade é executado.

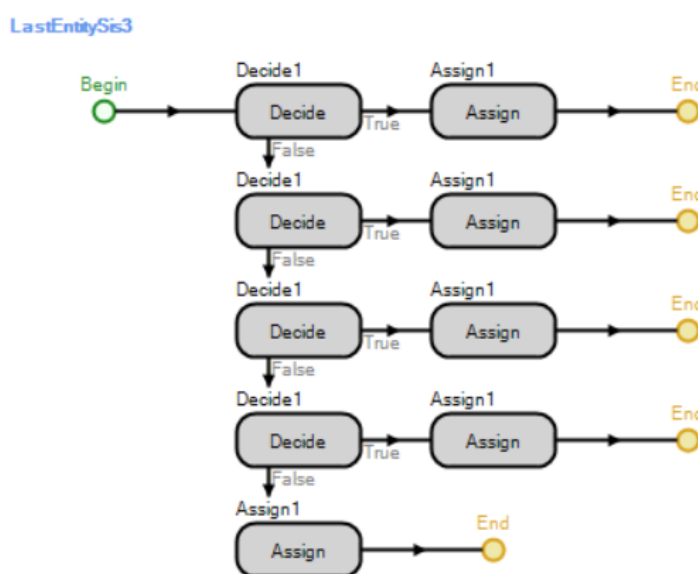


Figura 27: Identificação da ultima entidade processada

Existindo um processo para cada configuração de linha, a imagem corresponde ao processo que guarda a última entidade das workstations configuração de linha com reentrada. Facilitando a compreensão do processo uma vez que se trata da configuração com o menor número de workstations, somente involucrando 5 workstations. Através dos decides é identificado a workstation onde o processo foi executado. Sendo a

workstation identificada, o valor da entrada do vetor LastEnt3 correspondente a essa workstation toma o valor do IndexSetUp da entidade que acaba de ser processada. O processo é análogo para as restantes configurações de linha. Portanto, ao selecionar o tempo de *set up* a workstation utiliza a entrada do vetor LastEnt3 que lhe foi atribuída para selecionar a coluna da tabela de *set up*, onde o IndexSetUp da entidade que vai iniciar o processamento designa a linha da tabela. E uma vez finalizado o processamento o valor da entrada do vetor LastEnt3 é alterado antes de o tempo de *set up* ser aplicado para a seguinte entidade.

4.3.5 Cálculo de Estatísticas

Uma vez que uma entidade entra em uma Sink do modelo é executado um dos quatro processos análogos que realiza o cálculo de algumas estatísticas, ou seja, o cálculo do *flow time*, *lateness*, *tardiness*, número de trabalhos em atraso e adiantados.

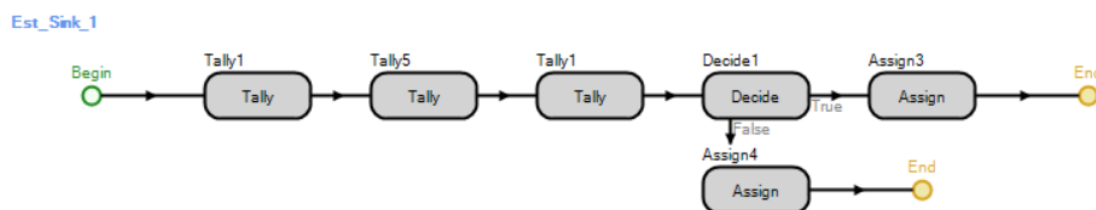


Figura 28: Cálculo de estatísticas

No processo segue-se a seguinte sequência de passos:

- *FowTime* - Calculado no primeiro *step* Tally é um elemento com unidade do tipo tempo, existindo um elemento correspondente para cada configuração de linha. O seu valor resulta da diferença entre o tempo de simulação e o *state* TempoEntradaSistema da entidade que ativou o processo.
- *Lateness* - Calculado no segundo *step* é um elemento com a mesma definição que *FlowTime*. Da mesma forma existe um elemento *Lateness* para cada configuração de linha. Toma o valor da diferença entre o tempo de simulação e o *state* DueDate.
- *Tardiness*- Calculado no último Tally é o máximo entre o *lateness* de cada entidade e o. Ainda que a média de este elemento, assim como a de todos os elemen-

tos, é incluída automaticamente no relatório dos experimentos, esta medida de desempenho não é utilizada devido a problemas de precisão.

- Decide - Comprova se o tempo de simulação é superior ao DueDate da entidade.
- Número atrasados - Caso o tempo de simulação for superior ao *state* DueDate, o valor do *state* N_atrasos aumenta em 1.
- Número adiantados - Se o tempo de simulação for igual ou inferior ao DueDate da entidade o valor do *state* N_adiantados aumenta em 1.

4.4 FORMULAÇÃO DAS REGRAS DE DESPACHO

Foram implementadas quatro regras de despacho, uma adaptação da regra Shortest Processing Time(SPT), Longest Process Sequence(LPS), Shortest Total Processing Time(STPT) e Earliest Due Date(EDD). Devido que a atribuição das prioridades é feita na pool antes das entidades entrarem o espaço fabril, a única regra de despacho dinâmica é a regra EDD com as restantes regras sendo estáticas. As excreções de todas as regras de despacho foram combinadas numa única formula, em que temos:

$$\begin{aligned} & \text{FatorProcessamento} \times \text{indexSPT} + (100 - \text{NOperacoes}) \times \text{IndexSPS} + \\ & (\text{MediaTempoProcessamento} \times \text{FatorProcessamento}) \times \text{IndexTPT} + \text{DueDate} \times \\ & \text{IndexEDD} \end{aligned}$$

- A regra SPT trata-se de uma adaptação devido à utilização do fator de processamento para indicar a prioridade das entidades. Sendo que a regra SPT utiliza o tempo de processamento de um determinado centro de trabalho, não se podendo ser usada diretamente ao se aplicar as regras de despacho nas pools.
- Para a regra LPS a entidade com maior número de operações necessários para finalizar a seu processamento recebe a maior prioridade. A sua formulação é feita ao inverter a regra Shortest Processing Sequence, este tipo de formulação provém do facto de as pools serem configuradas para libertar primeiro as entidades com o menor valor de prioridade.

- A MediaTempoProcessamento foi calculado e incluído na tabela Entidades para a aplicação da regra STPT, onde as entidades com o menor tempo total de processamento recebem a maior prioridade. Neste caso é considerado a média do tempo de processamento das workstations, media da distribuição uniforme, que executam os passos da sequência da entidade. Não foi incluído o fator de processamento no cálculo deste valor.
- O Due Date atribuído a cada entidade é utilizado pela regra EDD, sendo a maior prioridade atribuída ao menor valor de Due Date.

Ao se utilizar o processo RuleSelection somente um dos valores das regras de despacho é atribuído as entidades. E seguindo a formulação descrita independentemente da regra de despacho aplicada a entidade com o menor valor de priority tem a maior prioridade, podendo-se assim definir nas pools para ordenar as entidades pelo menor valor de priority. Assim de uma forma simples é possível permutar entre quatro regras de despacho, sem alterar o modelo de simulação.

4.5 VERIFICAÇÃO DOS COMPONENTES

Antes de executar qualquer experimento é necessário assegurar que o modelo realiza as atividades na forma pretendida. Para isto, após a elaboração do modelo foi criado um modelo de teste. Esta escolha deve-se a que para a verificação dos diversos detalhes do modelo é necessário fazer alterações no mesmo, assim sendo, se esta verificação fosse realizada no modelo desenvolvido e se cometesse algum erro em restaurar algum valor seria necessário voltar a realizar a verificação ou caso não fosse detetado resultaria num modelo errado.

A verificação deu início comprovando que cada elemento do modelo está definido corretamente, analisando-se os campos necessários de cada source, server, workstation e sink estão definidos corretamente.

Foram utilizados rótulos ligados a cada entidade do sistema para verificar os valores dos *states*, valores atribuídos a cada entidade na entrada do espaço fabril, assim como os valores de prioridade atribuídos. Os valores de prioridade são testados para assegurar

que os seus valores se alteram e são corretos ao modificar a variável de controlo das regras de despacho. É realizado o mesmo processo para verificar as variáveis de controlo restantes, nomeadamente o tempo na pool e percentagem de *set up*.

Para comprovar que o processo que controla que as entidades sejam libertadas para o espaço fabril no tempo de simulação correto, foram testados diversos valores para a variável de controlo executando o modelo a uma velocidade reduzida.

A sequência de trabalho, tempo de processamento e *set up* de cada centro de trabalho foram examinados ao modificar o modelo para libertar somente um tipo de entidade e ao fazer o seguimento da mesma. Foi criado um processo para medir o tempo despendido entre o início de processamento e a sua finalização. Posteriormente, o mesmo processo é utilizado para analisar o tempo de *set up*. No transcorrer deste processo foram verificados os componentes que participam na definição destes tempos.

Na última fase, rótulos foram utilizados para comprovar o processo que guarda a última entidade processada em cada centro de trabalho. Continuando com a verificação do processo que realiza o cálculo das estatísticas através do mesmo método. Ao seguir este procedimento é assegurado que o modelo representa o problema descrito no capítulo anterior, possibilitando uma correta análise do caso em estudo após a execução dos experimentos.

4.6 ANÁLISE DE PRECISÃO

Medidas de desempenho precisas são necessárias para a utilização correta da simulação. Para isto é necessário selecionar o número de replicações corretas de forma a minimizar o desperdício de esforço garantindo a precisão dos resultados. A precisão é perdida se o número de replicações for insuficiente, caso contrário um número de replicações demasiado elevado implica perda de tempo desnecessária.

Assim, depois de finalizar a execução dos experimentos definidos com 10 replicações é feita a extração de dados para Excel. Em Excel são realizados os cálculos para obter o número de replicações necessárias para lograr uma precisão de 5%. Podemos representar esta fórmula por:

$$n \cong n_0 \frac{h_0^2}{h^2}$$

n_0 é o numero de replicações iniciais, ou seja, o numero de replicações utilizadas para obter os dados sobre os quais são realizados os cálculos. h_0 é o *half width* das n_0 replicações iniciais

Ao utilizar esta fórmula é obtida uma aproximação do número de replicações necessárias para lograr a precisão desejada. h_0 e n_0 são conhecidos depois de correr a simulação com um determinado número de replicações. Com a anterior definição de que h tem que ser inferior a 5% do valor da média dos diversos indicadores de desempenho é feito o cálculo do número de replicações que permitem que os resultados tenham *half width* igual ou inferior a 5% da média. É selecionado o número de replicações mais elevado entre os valores obtidos.

Quanto menor for o *half width* mais precisos são os resultados, porem dependendo dos indicadores de desempenho pode não valer a pena o esforço computacional necessário para obter os resultados. Isto levou a uma menor precisão para alguns dos indicadores de desempenho, os quais possuem uma precisão entre 6% e 9%. Estes indicadores representam entre 1% e 3% dos resultados obtidos. Lograr uma precisão de 5% nestes indicadores implicava no mínimo o dobro de esforço computacional. Sendo pela mesma razão que algumas medidas de desempenho foram substituídas, como a percentagem de trabalhos atrasados pela de adiantados. Sendo que para estas medidas era necessário um enorme esforço computacional para obter a precisão necessária nos seus valores.

4.7 IMPLEMENTAÇÃO DO METAMODELO

O metamodelo utilizado é do tipo Kriging, a escolha da metodologia foi feita em base ao estudo realizado por Mohammed Reza Kianifar e Felician Campean (Kianifar and Campean, 2020), em que comparam vários metamodelos para aplicações nas áreas de engenharia. Para determinar a forma do metamodelo Kriging é necessário selecionar a família de correlação, o método de estimação dos *hyperparametros* e a *trend*. Em

relação a *trend* o metamodelo toma a forma de ordinary Kriging o qual foi discutido anteriormente.

Os metamodelos são implementados em UQLab. Um software de propósito geral para a quantificação de incerteza baseado em MatLab. Tem como foco a facilidade de utilização, personalização e manipulação de métodos estocásticos não intrusivos. Resultado da conexão de vários módulos científicos de livre acesso é permite implementar diversas metodologias como análise de sensibilidade, Monte Carlo, análise de confiabilidade e meatamodelagem. Sendo que suporta a metamodelação Kriging completamente personalizável, suportando tanto a interpolação como regressão Kriging, só é necessário definir os componentes do metamodelo.

Ainda que no documento de referência (Kianifar and Campean, 2020) é considerado respostas com ruído, o ruído é conhecido o que não é o caso no problema em estudo. Assim é optado pelo tipo de Kriging com melhor desempenho para maior diversidade de dados e por estimar a variância do ruído a partir dos dados para aplicar a regressão, uma funcionalidade do software UQLab.

4.7.1 Matérn

São funções de correlação estacionárias que dependem unicamente na posição relativa dos seus dois *inputs*. A família de correlação é de dimensão 1, por sua vez definida para um par de *inputs* de dimensão um $x, x' \in \mathbb{R}$ e parametrizado por $\theta \in \mathbb{R}_{>0}$.

A forma general da função de correlação Matérn é dada por

$$R(x, x'; \theta, v) = \frac{1}{2^{v-1} \Gamma(v)} \left(2\sqrt{v} \frac{|x - x'|}{\theta} \right)^v K_v \left(2\sqrt{v} \frac{|x - x'|}{\theta} \right)$$

onde θ é o parâmetro *scale* da função de correlação, $v \geq 1/2$, Γ é a função Gamma de Euler e K_v é a função de Bessel modificada do segundo tipo.

A função Matérn pode ser computada através de fórmulas simples quando $v = p + 1/2$, onde p é um inteiro não negativo (Rasmussen and Williams, 2006). Sendo as funções Matérn com $v = 3/2$ e $v = 5/2$ as mais utilizadas. Ainda que Matérn $3/2$ pode ter um melhor desempenho para alguns conjuntos de dados com a presença de ruído

(Kianifar and Campean, 2020), a função Matérn 5/2 é robusta e com melhor precisão para a maior diversidade de dados. Tem-se assim $v = 5/2$ para os metamodelos implementados, com a sua expressão sendo:

$$R(x, x'; \theta, v = 5/2) = \left(1 + \sqrt{5} \frac{|x - x'|}{\theta} + \frac{5}{3} \left(\frac{|x - x'|}{\theta}\right)^2\right) \exp \left[-\sqrt{5} \frac{|x - x'|}{\theta}\right]$$

4.7.2 Maximum-likelihood

O objetivo do método de estimação Maximum-likelihood é encontrar o conjunto de parâmetros β, σ^2, θ , e se aplicável, σ_n^2 de tal forma que a probabilidade dos *outputs* $Y = \{M(x^{(1)}), \dots, M(x^{(N)})\}^T$ é maximizada. Sendo que se supõe que Y segue distribuição normal multivariada, seguindo a suposição básica do Kriging, a função *likelihood* $L(\beta, \sigma, \theta; Y)$ expressa-se como

$$L(\beta, \sigma^2, \theta; Y) = \frac{(\det C)^{-1/2}}{(2\pi)^{N/2}} \exp \left[-\frac{1}{2} (Y - F\beta)^T C^{-1} (Y - F\beta) \right]$$

Onde a matriz de covariância C faz o sumário da matriz do processo Gaussiano $\sigma^2 R$ e a matriz de covariância dos ruídos dos *outputs* Σ_n

$$C = \sigma^2 R + \Sigma_n$$

Dependendo se o ruído está presente e da natureza do mesmo, diferentes problemas de otimização podem ser definidos. No caso de *outputs* com ruído com uma variância do ruído desconhecida homogênea ou homocedastica, a matriz de covariância C reduz-se a $\sigma_{total}^2 \tilde{R}$ em que, como introduzido anteriormente,

$$\sigma_{total}^2 = \sigma^2 + \sigma_n^2$$

$$\tilde{R} = (1 - \tau)R + rI$$

$$\tau = \frac{\sigma_n^2}{\sigma_{total}^2}$$

o *likelihood* correspondente lê-se

$$L(\beta, \sigma^2, \theta, \tau; Y) = \frac{(\det \tilde{R})^{-1/2}}{(2\pi\sigma_{total}^2)^{N/2}} \exp \left[-\frac{1}{2\sigma_{total}^2} (Y - F\beta)^T \tilde{R}^{-1} (Y - F\beta) \right]$$

Ao maximizar a quantidade descrita nesta equação, as seguintes estimativas analíticas de β e σ_{total}^2 sendo estritamente funções de θ são obtidas (Dubourg, 2011)

$$\hat{\beta} = \beta(\theta) = (F^T \tilde{R}^{-1} F)^{-1} F^T \tilde{R}^{-1} Y$$

$$\hat{\sigma}_{total}^2 = \sigma_{total}^2(\theta) = \frac{1}{N} (Y - F\beta)^T \tilde{R}^{-1} (Y - F\beta)$$

Seguindo estas equações, podemos expressar o problema de otimização na forma

$$\hat{\theta}, \hat{\tau} = \arg \min_{\theta \in D_\theta, \tau \in (0,1)} \frac{1}{2} \left[\log(\det \tilde{R}) + N \log(2\pi\hat{\sigma}_{total}^2) + N \right]$$

Note-se que em adição ao parâmetro θ é necessário otimizar o parâmetro τ estando delimitado em $(0, 1)$.

4.7.3 Método de Otimização

Para resolver o problema de otimização existem *trade-offs* entre a escolha de alguns métodos locais ou globais. Métodos locais tendem a convergir mais rapidamente e exigem menos avaliações das funções objetivo, porem podem ter uma má performance devido à existência de regiões planas e múltiplos mínimos locais, especialmente quando a dimensão de *inputs* aumenta. Alternativamente, podem ser aplicados os métodos híbridos que combinam os aspetos dos dois métodos.

O método aplicado é a combinação L-BFGS, em que um método baseado no gradiente de ponto interior é usado para aproximar a matriz Hessiana usando uma variante de memória limitado do método *quasi-Newton Broyden-Fletcher-Goldfarb-Shanno* (L-BFGS) (Byrd et al., 1999) e Algoritmo genético, de uma forma simples é um procedimento probabilístico destinado a ambientes que podem ser representados por cordas. Um método estando bem estabelecido e é aplicado em diversas áreas (Goldberg, 1989). O

método implementado é uma abordagem híbrida onde a solução final do algoritmo genético é utilizado como ponto de partida do método baseado em gradiente.

DISCUSSÃO DE RESULTADOS

O trabalho realizado centrou-se no estudo de uma família de produtos e na análise do modo como o seu processamento é afetado pela configuração do sistema de produção. O objetivo foi determinar as ligações entre as variáveis e o desempenho do sistema de produção.

A abordagem baseia-se na elaboração de vários metamodelos em que se pode observar os valores obtidos para as várias medidas de desempenho, onde o tempo de *set up* é utilizado para obter os pontos necessários para a metamodelagem. Ao serem considerados vários cenários em que são colocados os valores de todas as variáveis, são obtidos vários metamodelos onde é possível observar as consequências da alteração dos valores de uma variável. O método baseado em metamodelos permitiu manipular a grande quantidade de informação num reduzido intervalo de tempo.

O estudo desta dissertação baseia-se num modelo teórico criado para este trabalho, onde foram realizados diversos *trade-offs* com o objetivo de se obterem resultados significativos para a maior quantidade possível de medidas de desempenho, resultando na incapacidade de analisar a percentagem de produtos finalizados no prazo do sistema com máquinas paralelas, tendo sempre o valor 100%. Contudo, os metamodelos foram capazes de englobar as características gerais do sistema de forma eficaz, onde se pode observar as repercussões das alterações no modelo.

5.1 TEMPO NA POOL

O tempo na pool é a variável mais afetada pela necessidade de limitar a análise do trabalho, somente temos três pontos de análise. Embora não seja claro devido à carência de pontos de análise, existem algumas evidências do facto desta variável afetar o sistema.

Com um aumento do tempo na pool dá-se um aumento dos valores do sistema com máquinas paralelas. O número de produtos finalizados diminui, no entanto, devido à existência de vários mínimos e máximos locais, é possível obter melhores valores para tempos na pool elevados numa determinada percentagem de *set up*.

Nas restantes configurações de linha o aumento do tempo na pool provoca um aumento dos valores para percentagens de *set up* baixas. O contrário ocorre para percentagens elevadas de *set up*, onde se obtêm melhores valores com o aumento do tempo na pool. A diferença dos valores para as percentagens baixas é reduzida e para as percentagens mais altas é considerável. Ainda, é possível observar em alguns casos que existe um ponto de inflexão onde o aumento do tempo na pool começa a resultar em piores resultados.

5.2 LATENESS

É importante realçar uma particularidade do comportamento das medidas de desempenho *lateness*: possuindo valores negativos, são preferíveis os valores elevados sendo os mais próximos de 0. Tal é preferível uma vez que os valores de *lateness* negativos implicam a manutenção de artigos em stock o que, num caso real, representa custos. Sendo assim, o aumento do tempo na pool resulta em melhores valores para as percentagens reduzidas de *set up* e piores valores para as percentagens elevadas.

5.3 REGRAS DE DESPACHO

A influência das regras de despacho no desempenho do sistema para percentagens reduzidas de tempo de *set up* é desprezível, podendo existir uma regra com um ligeiro

melhor desempenho, conforme a medida de desempenho e a configuração de linha. No entanto, com o aumento da percentagem do tempo de *set up* a regra de despacho SPT (Shortest Processing Time) tem o melhor desempenho, seguida da regra STPT (Shortest Total Processing Time). Para alguns cenários a regra STPT pode igualar ou superar o desempenho da regra SPT. A regra com pior desempenho é a regra EDD (Earliest Due Date), exceto para a medida de desempenho *lateness* onde proporciona os melhores valores.

5.4 CONFIGURAÇÕES DE LINHA

Com uma grande diferença, a linha com máquinas paralelas tem o melhor desempenho. Os valores da linha com máquinas paralelas seguem uma distribuição exponencial, exceto o número de produtos finalizados. Os valores das linhas restantes são lineares.

A linha simples e a linha com salto têm um desempenho similar, sendo as diferenças nos seus valores mínimos. A utilização das máquinas da linha simples diminui com o aumento da percentagem de *set up*, enquanto a das outras configurações de linha aumenta. Este é o aspeto que diferencia a linha simples da linha com salto.

5.5 TEMPO DE SIMULAÇÃO

Com a variação do tempo de simulação o desempenho das regras de despacho mantém-se, obtendo-se um melhor desempenho da regra SPT e o pior desempenho da regra EDD. Nas primeiras horas de processamento a diferença entre a linha simples, com reentrada e com salto é mínima. Diferenciando-se a linha com reentrada com o transcorrer do tempo da linha simples e da linha com salto. Os valores destas três linhas são lineares: os valores aumentam constantemente. O que não acontece com a linha com máquinas paralelas: os valores aumentam no início da simulação e depois mantém-se constante. Ainda para a utilização das máquinas todas as linhas estabilizam num valor, portanto a utilização das máquinas das linhas mantém-se constante durante o processamento.

CONCLUSÃO E TRABALHO FUTURO

Ao utilizar a metamodelagem foi possível realizar uma análise de uma extensa quantidade de dados envolvendo diversas linhas e variáveis, num tempo relativamente curto. Apesar de um desempenho satisfatório da utilização de metamodelos para análise do modelo definido, a mesma aborda pouco detalhes de cada linha de produção. Sendo que, ao utilizar a percentagem de *set up* como o ponto central de estudo a análise do sistema vê-se limitado. Para uma melhor análise do sistema é necessário a elaboração dos metamodelos dependendo das restantes variáveis. No entanto, uma análise tão ampla do sistema resultaria num incremento ao número já elevado de metamodelos incluídos neste trabalho. Abordando cada configuração de linha por separado é possível analisar cenários mais detalhados, mantendo uma quantidade de informação fácil de manipular através da metodologia implementada. Podendo indicar novos comportamentos do modelo.

Ainda para ser possível a comparação dos metamodelos é utilizada a mesma metodologia para criar os metamodelos, no entanto, não se adequa para todos os cenários abordados. Como é o caso do número de produtos finalizados da linha com máquinas paralelas, tendo os dados dispersos com metamodelos lineares. A metodologia também pode ser ajustada a cada caso específico, melhorando os metamodelos obtidos.

Contudo, apesar de todas as limitações assinaladas este trabalho é uma boa base para trabalhos futuros. É explorado a área da metamodelagem de uma forma progressiva, fundamentando as bases das mesmas. A informação é apresentada de uma forma simples, facilitando a sua compressão para pessoas não familiarizadas com a área. Um fato extremamente importante no ponto de tempo em que foi realizado este trabalho,

uma vez que é difícil encontrar documentação que incida nas bases da metamodelação. Sendo necessário consultar diversos fontes para obter a informação necessária.

É demonstrado que através da implementação desta metodologia é possível analisar e identificar relações entre as variáveis de um sistema de produção, de uma forma relativamente simples. Sendo este trabalho uma boa introdução da metodologia. Um possível próximo passo seria a análise mais detalhada dos componentes do modelo proposto. Assim como, implementação de metodologias alternativas e verificar os resultados obtidos com casos reais.

BIBLIOGRAFIA

Al-Salem, A. (2004). A heuristic to minimize makespan in proportional parallel flow shops. *International Journal of Computing and Information Sciences*, 2(2), 98–107.

Altioek, T. and Melamed, B. (2007). *Simulation Modeling and Analysis with ARENA*. Academic Press.

Ammeri, A., Hachicha W., Masmoudi, F. and Habib Chabchoub, H. (2011). A comprehensive literature review of mono-objective simulation optimization methods. *Advances in Production Engineering and Management*, 6, 291–302.

Ankenman, B., Nelson, B.L., and Staum, J. (2008). Stochastic kriging for simulation metamodeling. In *Winter Simulation Conference*.

April, J., Glover, F., Laguna, M. and, Kelly, J.P. (2003). Practical introduction to simulation optimization. In *Winter Simulation Conference*.

Azadeh, A., Moghaddam, M., Asadzadeh, S.M., and, Negahban, A. (2011). An integrated fuzzy simulation-fuzzy data envelopment analysis algorithm for job-shop layout optimization: The case of injection process with ambiguous data. *European Journal of Operational Research*, 214(3), 768–779.

Azadivar, A. (1992). A tutorial on simulation optimization. In *Winter Simulation Conference*.

Azadivar, F. and, Wang, J. (2000). Facility layout optimization using simulation and genetic algorithms. *International Journal of Production Research*, 38(17), 4369–4383.

Balaban, M. and, Dengiz, B.(2018). Lognormal ordinary kriging metamodel in simulation optimization. *Operations Research and Applications : An International Journal*, 5(1):1–12.

Barton R.R. and, Meckesheimer, M.(2006). Metamodel-based simulation optimization. *Handbooks in Operations Research and Management Science: Simulation*,13,535–574.

Barton R. R.(2009). Simulation optimization using metamodels. In *Winter Simulation Conference*,230–238.

Barton R. R.(2015). Tutorial: Simulation metamodeling. In *Winter Simulation Conference*.

Biles, W.E., Kleijnen, J. P.C., van Beers, W.and Van Nieuwenhuyse, I. (2007). Kriging metamodeling in constrained simulation optimization: An explorative study. In *Winter Simulation Conference*.

Blanning, R.W.(1974). The sources and uses of sensitivity information. *Interfaces*, 4(4), 32–38.

Blanning, R.W.(1975a). The construction and implementation of metamodels. *Simulation*, 24,177–184.

Blanning, R.W.(1975b). Response to michel, kleijnen and permut. *Interfaces*, 5(3),24–25.

Boris, J., Patnaik, G., Young Lee, M., Young, T., Leidl, B., Harms, F. and Schatzmann, M.(2009). Validation of an les urban aerodynamics model for homeland security. In *47th AIAA Aerospace Sciences Meeting*, 5–8.

Box, G.E. and Draper, N.R.(1987). *Empirical Model-Building and Response Surfaces*. John Wiley and Sons.

Byrd, R.H., Hribar, M.E. and Nocedal, J.(1999). An interior point algorithm for large scale nonlinear programming. *SIAM Journal on Optimization*, 9(4),877–900.

Can, B. and Heavey, C.(2012). A comparison of genetic programming and artificial neural networks in metamodeling of discrete-event simulation models. *Computers and Operations Research*, 39(2),424–436.

Carson, Y. and Maria, A.(1997). Simulation optimization: Methods and applications. In Winter Simulation Conference, 118–126.

Chaudhry, I.A. and Khan, A.(2016). A research survey: Review of flexible job shop scheduling techniques. *International Transactions in Operational Research*, 23(3),551–591.

Chen, L., Wang, H., Ye, F. and Hu, W.(2019). Comparative study of HDMRS and other popular metamodeling techniques for high dimensional problems. *Structural and Multidisciplinary Optimization*, 59(1),21–42.

Chen, X. and Kim, K.(2013). Building metamodels for quantile-based measures using sectioning. In Winter Simulation Conference.

Chen, X., Yang, H., and Sun, K.(2017). Developing a meta-model for sensitivity analyses and prediction of building performance for passively designed high-rise residential buildings. *Applied Energy*, 194,422–439.

Cheng, J., Karuno, Y. and Kise, H.(2001). A shifting bottleneck approach for a parallel-machine flowshop scheduling problem. *Journal of the Operations Research*, 44 (2),140–156.

Cheng, R. and Currie, C.S.M.(2004). Optimization by simulation metamodeling methods. In Winter Simulation Conference.

Critchlow, M., Dodd, K., Chou, J. and van de Hoek, A.(2003). Refactoring product line architectures. In First International Workshop on Refactoring: Achievements, Challenges, and Effects (REFACE'03),23–26.

De la Fuente, R. and Smith III, R.(2017). Metamodeling a system dynamics model: A contemporary comparison of methods. In Winter Simulation Conference, 1926–1937.

Dellino, G., Kleijnen, J.P.C. and Meloni, C.(2009). Robust simulation-optimization using metamodels. In Winter Simulation Conference.

Diamond, B., Krah, D., Nastasi, A. and Tag, P.(2010). Extendsim advanced technology: Integrated simulation database. In Winter Simulation Conference.

Dubourg, V.(2011). Adaptive surrogate models for reliability analysis and reliability-based design optimization. PhD thesis, Universite Blaise Pascal, Clermont-Ferrand, France.

Ducloux, R.(2014). Improvement of part or tooling life prediction through simulation of whole manufacturing process. *Procedia Engineering*, 81,504–509.

Fang, H. and Horstemeyer, M.F.(2006). Global response approximation with radial basis functions. *Engineering Optimization*, 38(4), 407–424.

Figueira, G. and Almada-Lobo, B.(2014). Hybrid simulation–optimization methods: A taxonomy and discussion. *Simulation Modelling Practice and Theory*, 46, 118–134.

Forrester, A., Sobester, A. and Keane, A.(2008). *Engineering Design via Surrogate Modelling: A Practical Guide*. John Wiley and Sons.

Fu, M.C.(2001). Simulation optimization. In *Winter Simulation Conference*, 53–61.

Fu, M.C., Glover, F. and April, J.(2005). Simulation optimization: A review, new developments, and applications. In *Winter Simulation Conference*.

Gapp, W., Mankekar, P.S. and Mitten, L.G.(1965). Sequencing operations to minimize in-process inventory costs. *Management Science*, 11(3), 476–484.

Garcia, E.F., Zuniga, E.R., Bruch, J., Moris, M.U. and Syberfeldt, A.(2018). Simulation-based optimization for facility layout design in conditions of high uncertainty. In *51st CIRP Conference on Manufacturing Systems*, 334–339.

Garey, M.R., Johnson, D. S. and Sethi, R.(1976a). The complexity of flowshop and jobshop scheduling. *Mathematics of Operations Research*, 1(2), 117–129.

Garey, M.R., Graham, R.L. and Johnson, D.S.(1976b). Some np-complete geometric problems. In *STOC '76: Proceedings of the eighth annual ACM symposium on Theory of computing*, 10–22.

Gere, W.S.(1966) Heuristics in job shop scheduling. *Management Science*, 13(3), 167–190.

Goldberg, D.E.(1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Professional.

Gonçalves, J., Mendes, J. and Resende, M.(2005). A hybrid genetic algorithm for the job shop scheduling problem. *European Journal of Operational Research*, 167(1), 77–95.

Gyulai, D., Szaller, A. and Viharos, Z.J.(2016). Simulation-based flexible layout planning considering stochastic effects. In 49th CIRP Conference on Manufacturing Systems (CIRP-CMS 2016), 177–182.

Haupt, R.(1989). A survey of priority rule-based scheduling. *Operations-Research-Spektrum*, 11 (1), 3–16.

Heath, S.K., Bard, J.F. and Morrice, D.J.(2013). A grasp for simultaneously assigning and sequencing product families on flexible assembly lines. *Annals of Operations Research*, 203(1), 295–323.

Heilala, J., Montonen, J., Salmela, A. and Jarvenpaa,P.(2007). Modeling and simulation for customer driven manufacturing system design and operations planning. In Winter Simulation Conference.

Hester, J., Gregory, J. and Kirchain, R.(2017). Sequential early-design guidance for residential single-family buildings using a probabilistic metamodel of energy consumption. *Energy and Buildings*, 134,202–211.

Jagstam, M. and Klingstam, P.(2002). A handbook for integrating discrete event simulation as an aid in conceptual design of manufacturing systems. In Winter Simulation Conference, 1940–1944.

Jithavech, I. and Krishnan, K.(2010). A simulation-based approach for risk assessment of facility layout designs under stochastic product demands. *International Journal of Advanced Manufacturing Technology*, 49(1), 27–40.

Kaban, A.K., Othman, Z. and Rohmah, D.(2012). Comparison of dispatching rules in job-shop schedulingproblem usingsimulation: A case study. *International Journal of Simulation Modelling*, 11(3), 129–140.

Kapanoglu, M. and Alikalfa, M.(2011). Learning if-then priority rules for dynamic job shops using genetic algorithms. *Robotics and Computer-Integrated Manufacturing*, 27(1), 47–55.

Kianifar, M.R. and Campean, F.(2020). Performance evaluation of metamodeling methods for engineering problems: towards a practitioner guide. *Structural and Multidisciplinary Optimization*, 61(1), 159–186.

Kim, B., Lee, Y., and Choi, D.(2009). Comparison study on the accuracy of metamodeling technique for non-convex functions. *Journal of Mechanical Science and Technology*, 23(4), 1175–1181.

Kleijnen, J.P.C.(1975). A comment on blanning's metamodel for sensitivity analysis: the regression metamodel in simulation. *Interfaces*, 5(3), 21–23.

Kleijnen, J.P.C.(2007). Regression models and experimental designs: a tutorial for simulation analysts. In *Winter Simulation Conference*.

Krahl, D.(2009). Extendsim advanced technology: Discrete rate simulation. In *Winter Simulation Conference*.

Krige, D.G.(1951). A statistical approach to some mine valuation and allied problems on the witwatersrand. Master's thesis, University of the Witwatersrand, South Africa.

Kroetz, H.M., Tessari, R.K. and Beck, A.T.(2017). Performance of global metamodeling techniques in solution of structural reliability problems. *Advances in Engineering Software*, 114, 394–404.

Law, A.M.(2015). *Simulation Modeling and Analysis*. McGraw-Hill Education, 5th edition.

Li, Y., Ng, S.H., Xie, M. and Goh, T.N.(2010). A systematic comparison of metamodeling techniques for simulation optimization in decision support systems. *Applied Soft Computing*, 10(4), 1257–1273.

Liu, H., Xu, S. and Wang, X.(2016). Sampling strategies and metamodeling techniques for engineering design: Comparison and application. In *ASME Turbo Expo 2016: Turbine Technical Conference and Exposition*.

Mendes, A.R., Ferreira Andrade Ramos, A.L., Simaria, A.S. and Vilarinho, P.(2005). Combining heuristic procedures and simulation models for balancing a pc camera assembly line. *Computers and Industrial Engineering*, 49(3), 413–431.

Michalos, G., Makris, S. and Mourtzis, D.(2012). An intelligent search algorithmbased method to derive assembly line design alternatives. *International Journal of Computer Integrated Manufacturing*, 25(3), 211–229.

Moon, D.H., Shin, K.W., Park, C.S. and Kim, D.S.(2009). A manufacturing system simulation of semiconductor packaging substrate. In *Spring Simulation Multiconference*, 1391–1396.

Moslemipour, G., Lee, T.S. and Rilling, D.(2012). A review of inteligente approaches for designing dynamic and robust layouts in flexible manufacturing systems. *International Journal of Advanced Manufacturing Technology*, 60(1), 11–27.

Negahban, A. and Smith, J.S.(2014). Simulation for manufacturing system design and operation: Literature review and analysis. *Journal of Manufacturing Systems*, 33(2), 241–261.

Pan, J.H. and Chen, J.S.(2003). Minimizing makespan in re-entrant permutation flow-shops. *Journal of the Operational Research Society*, 54(6), 642–653.

Pegden, C.D.(2007). Simio: A new simulation system based on intelligent objects. In *Winter Simulation Conference*.

Pegden, C.D. and Sturrock, D.T.(2011). Introduction to simio. In *Winter Simulation Conference*.

Pezzella, F., Morganti, G. and Ciaschetti, G.(2008). A genetic algorithm for the flexible job-shop scheduling problem. *Computers and Operations Research*, 35(10), 3202–3212.

Qiu, X. and Lau, H.Y.K.(2013). An ais-based hybrid algorithm with pdrs for multiobjective dynamic online job shop scheduling problem. *Applied Soft Computing*, 13(3), 1340–1351.

Rajabinasab, A. and Mansour, S.(2011). Dynamic flexible job shop scheduling with alternative process plans: an agent-based approach. *The International Journal of Advanced Manufacturing Technology*, 54, 1091–1107.

Rasmussen, C. and Williams, C.(2006). *Gaussian Processes for Machine Learning. Adaptive Computation and Machine Learning*. MIT Press, Cambridge, Massachusetts.

Rossi, A.(2014). Flexible job shop scheduling with sequence-dependent setup and transportation times by ant colony with reinforced pheromone relationships. *International Journal of Production Economics*, 153(4), 253–267.

Ruiz, R. and Vázquez Rodríguez, J.A.(2010). The hybrid flow shop scheduling problem. *European Journal of Operational Research*, 205(1), 1–18.

Sacks, J., Welch, W.J., Mitchell, T.J. and Wynn, H.P.(1989). Design and analysis of computer experiments. *Statistical Science*, 4(4), 409–423.

Salemi, P., Nelson, B.L. and Staum, J.(2012). Moving least squares regression for high dimensional simulation metamodeling. In *Winter Simulation Conference*.

Santner, T.J., Williams, B.J. and Notz, W.I.(2003). *The Design and Analysis of Computer Experiments*. Springer Series in Statistics. Springer.

Sardar, V., Polshettiwar, A., Trivedi, D. and Rajhans, N.R.(2016). Optimization of plant layout using simulation software. In *Design and Research Conference (AIMTDR-2016)*.

Shanthikumar, J. and Sargent, R.(1983). A unifying view of hybrid simulation/analytic models and modeling. *Operations Research*, 31, 1030–1052.

Song, J., Lei, W., Bei, N., Zavala, M., Foy, B., Volkamer, R., Cardenas, B., Zheng, J., Zhang, R. and Molina, L.T.(2010). Ozone response to emission changes: a modeling study during the 30 mcma-2006/milagro campaign. *Atmospheric Chemistry and Physics*, 10, 3827–3846.

Østergard, T., Jensen, R.L. and Maagaard, S.E.(2018). A comparison of six metamodeling techniques applied to building performance simulations. *Applied Energy*, 211, 89–103.

- Sturrock, D.T. and Pegden, C.D.(2010). Recent innovations in simio. In Winter Simulation Conference.
- Tekin, E. and Sabuncuoglu, I.(2004). Simulation optimization: A comprehensive review on theory and applications. *IIE Transactions*, 36(11),1067–1081.
- Terzi, S. and Cavalieri, S.(2004). Simulation in the supply chain context: a survey. *Computers in Industry*, 53(1), 3–16.
- Ullman, D.G.(2017). The mechanical design process. McGraw-Hill, 6th edition.
- Van Gelder, L., Das, P., Janssen, H. and Roels, S.(2014a). Comparative study of metamodeling techniques in building energy simulation: guidelines for practitioners. *Simulation Modelling Practice and Theory*, 49, 245–257.
- Van Gelder, L., Janssen, H. and Roels, S.(2014b). Probabilistic design and analysis of building performances: Methodology and application example. *Energy and Buildings*, 79, 202–211.
- Villa-Vialaneix, N., Follador, M., Ratto, M. and Leip, A.(2012). A comparison of eight metamodeling techniques for the simulation of n₂o fluxes and n leaching from corn crops. *Environmental Modelling and Software*, 34, 51–66.
- Vinod, V. and Sridharan, R.(2008). Dynamic job-shop scheduling with sequencedependent setup times: simulation modeling and analysis. *The International Journal of Advanced Manufacturing Technology*, 36(3), 355–372.
- Wang, G.G. and Shan, S.(2007). Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical Design*, 129(4), 370–380.
- Wang, G., Yan, Y., Zhang, X. and Shangguan, J.(2008). A simulation optimization approach for facility layout problem. In *IEEE International Conference on Industrial Engineering and Engineering Management*, 734–738.
- Wang, H., Shan, S., Wang, G. and Li, G.(2011). Integrating least square support vector regression and mode pursuing sampling optimization for crashworthiness design. *Journal of Mechanical Design*, 133(4).

Xia, W. and Wu, Z.(2005). An effective hybrid optimization approach for multiobjective flexible job-shop scheduling problems. *Computers and Industrial Engineering*, 48(2), 409–425.

Xie, W., Nelson, B.L. and Barton, R.R.(2014). Statistical uncertainty analysis for stochastic simulation with dependent input models. In *Winter Simulation Conference*.

Yan, Y. and Wang, G.(2007). A job shop scheduling approach based on simulation optimization. In *2007 IEEE International Conference*, 1816–1822.

Yu, B. and Popplewell, K.(1994). Metamodels in manufacturing: a review. *International Journal of Production Research*, 32(4), 787–796.

Yuan, J., Nian, V., Su, B. and Meng, Q.(2017). A simultaneous calibration and parameter ranking method for building energy models. *Applied Energy*, 206, 657–666.

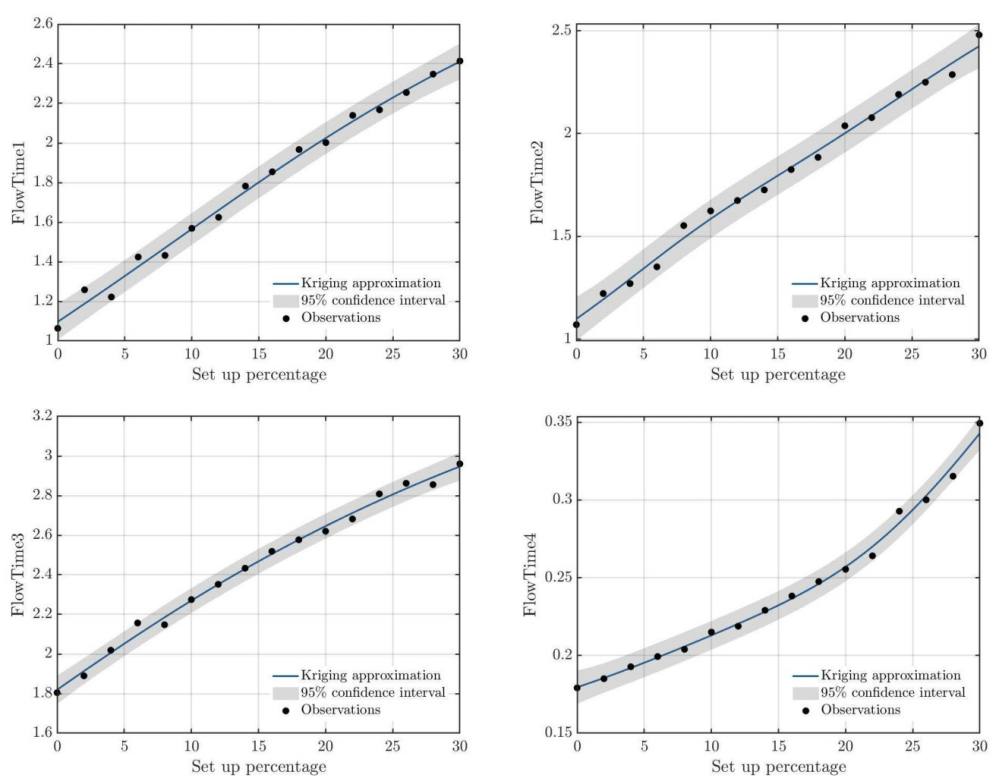
Zakerifar, M., Biles, W.E. and Evans, G.W.(2009). Kriging metamodeling in multi-objective simulation optimization. In *Winter Simulation Conference*.

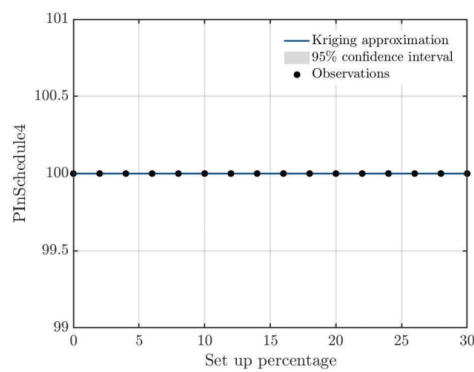
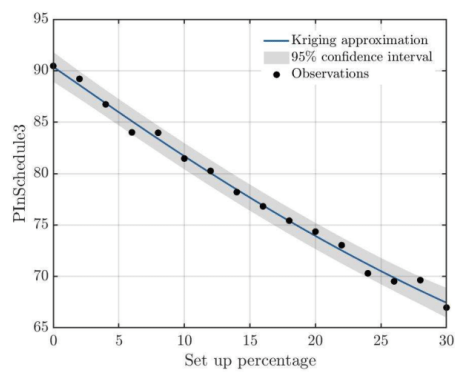
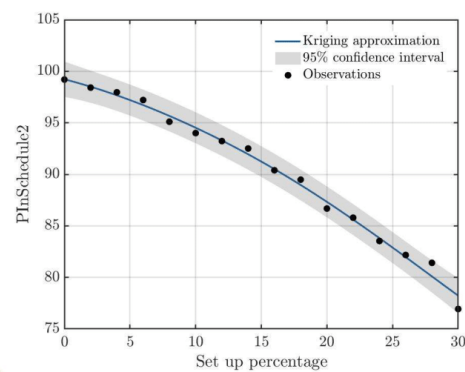
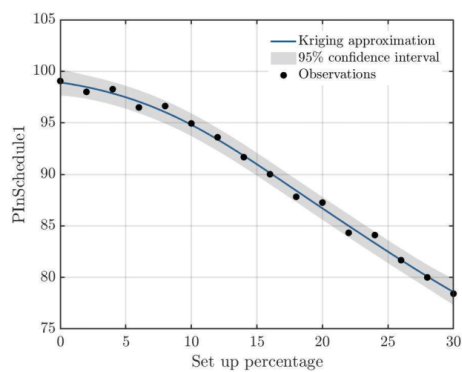
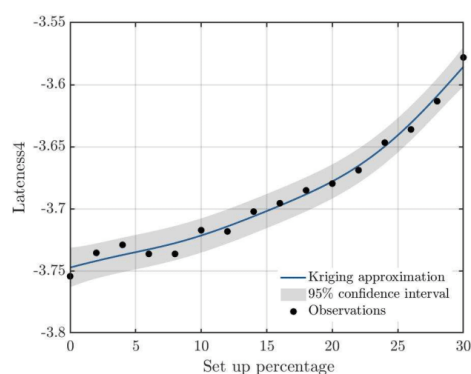
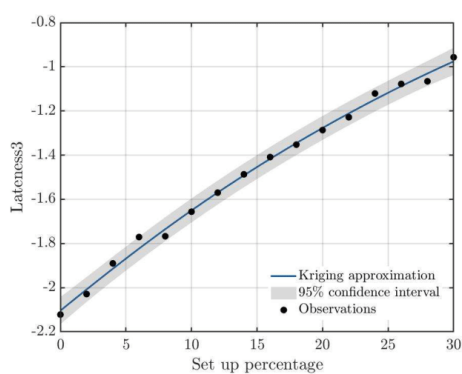
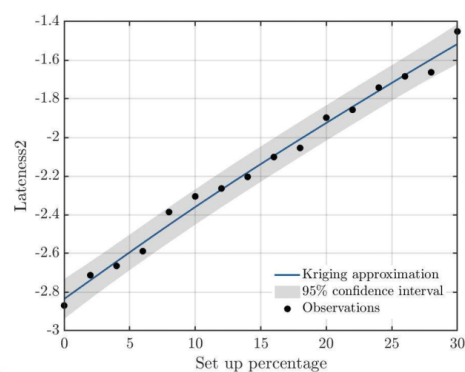
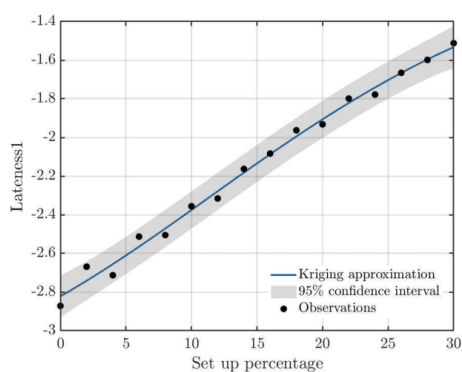
Zhang, Z., Wang, X., Wang, X., Cui, F. and Cheng, H.(2019). A simulationbased approach for plant layout design and production planning. *Journal of Ambient Intelligence and Humanized Computing*, 10(3), 1217–1230.

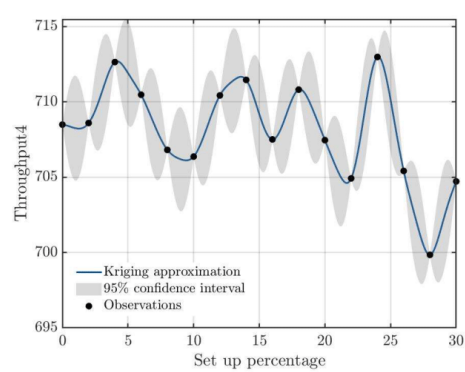
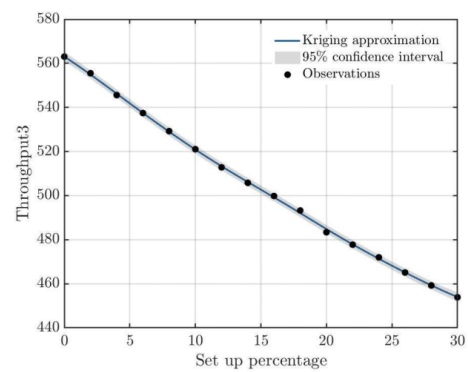
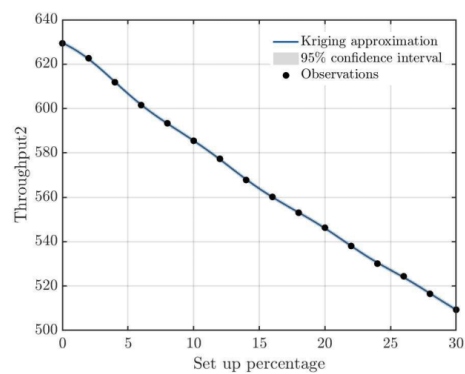
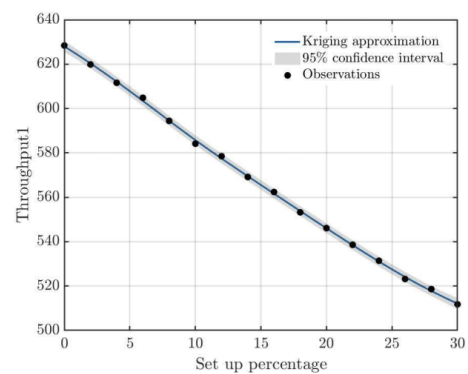
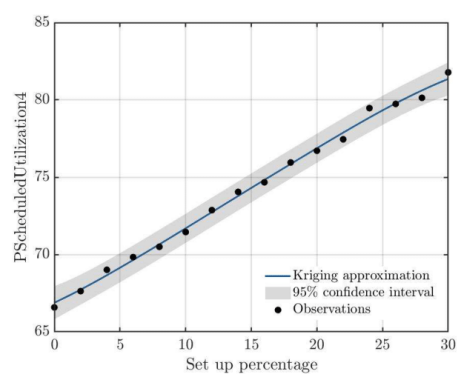
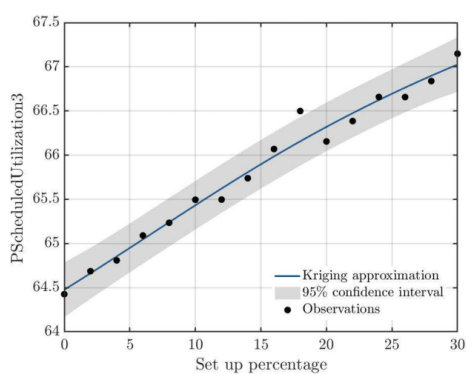
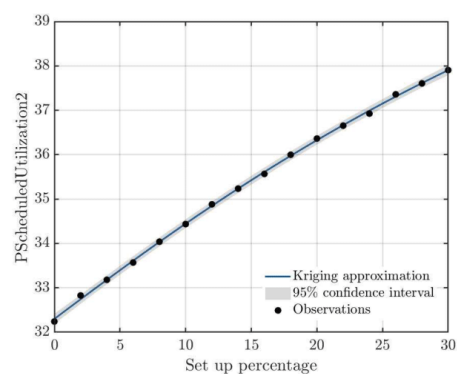
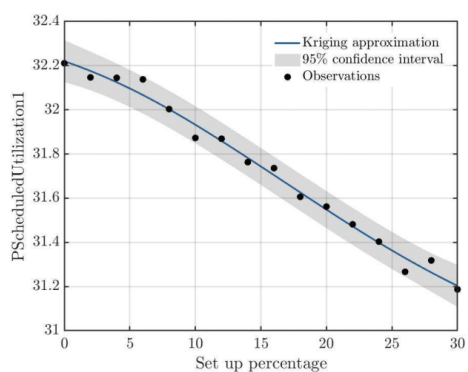
Zhao, D. and Xue, D.(2010). A comparative study of metamodeling methods considering sample quality merits. *Structural and Multidisciplinary Optimization*, 42(6), 923–938.

Zhao, W. and Bu, L.(2019). Global sensitivity analysis with a hierarchical sparse metamodeling method. *Mechanical Systems and Signal Processing*, 115, 769–781.

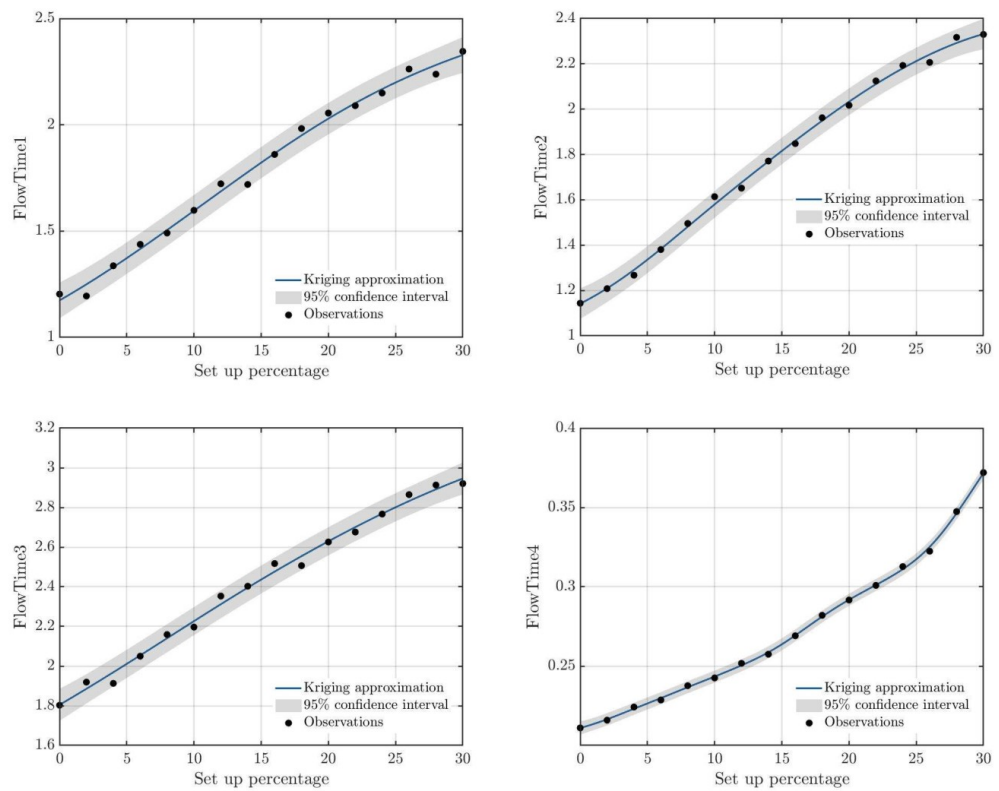
REGRA EARLIEST DUE DATE COM 5 MINUTOS NA POOL

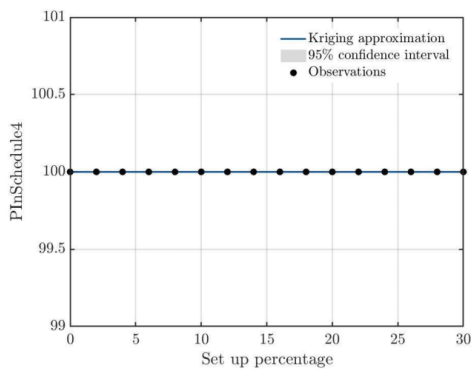
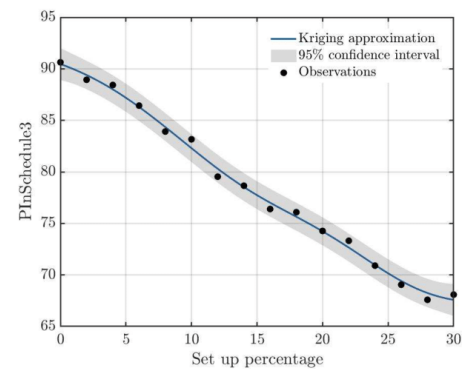
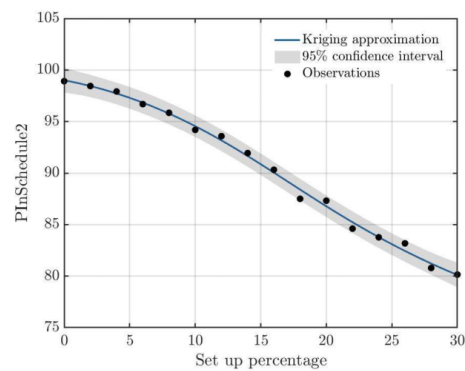
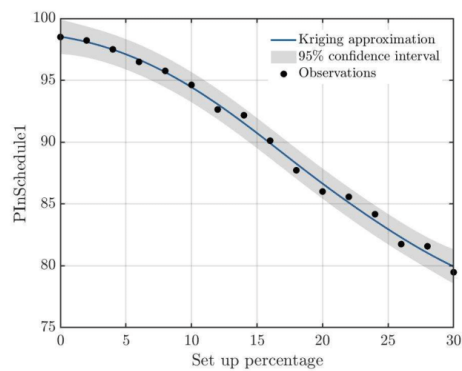
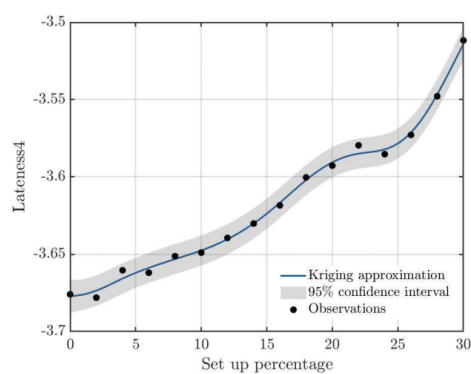
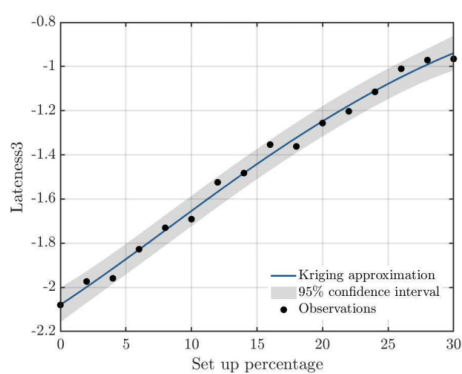
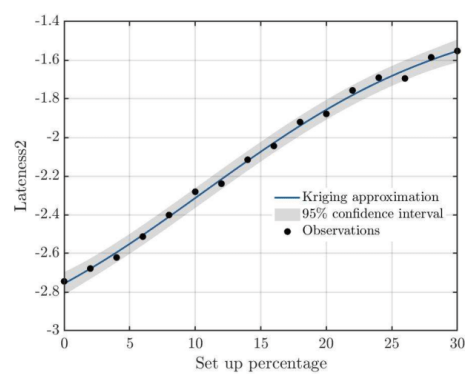
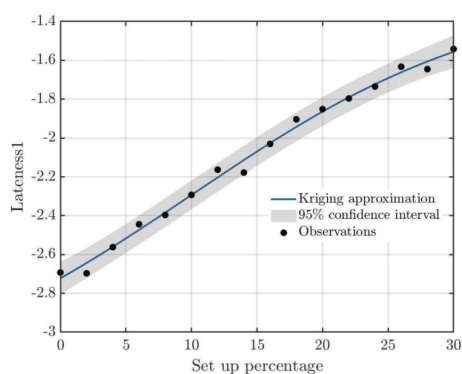


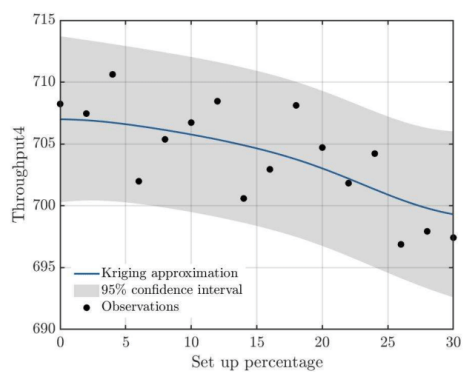
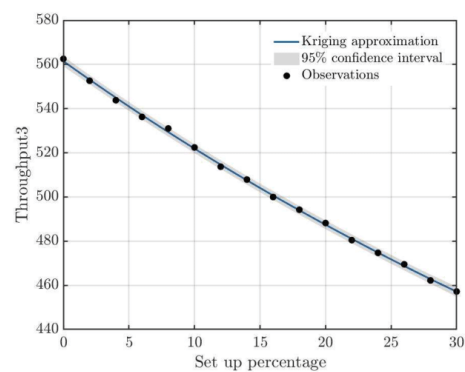
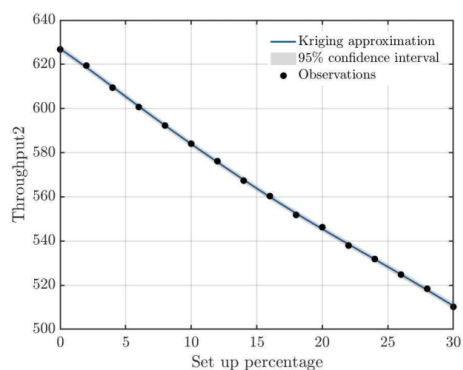
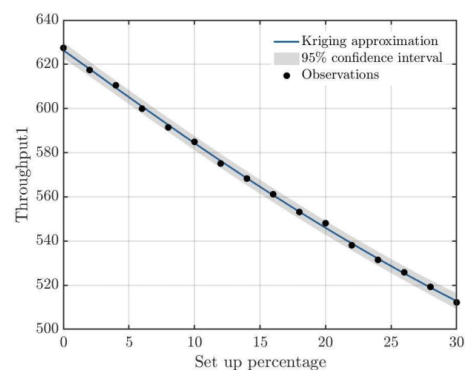
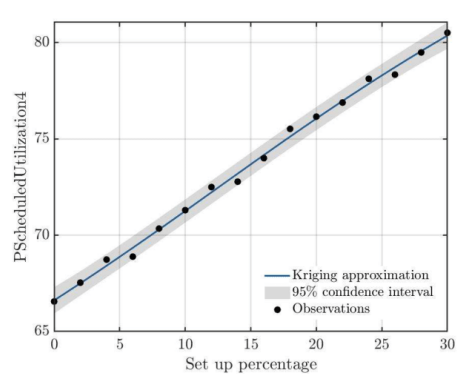
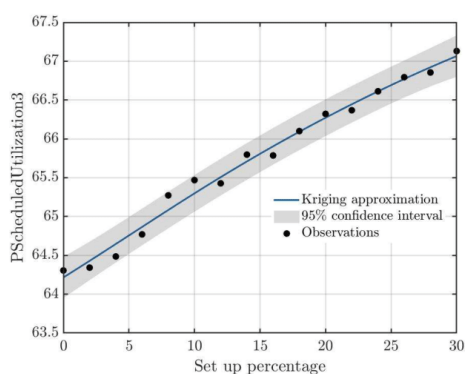
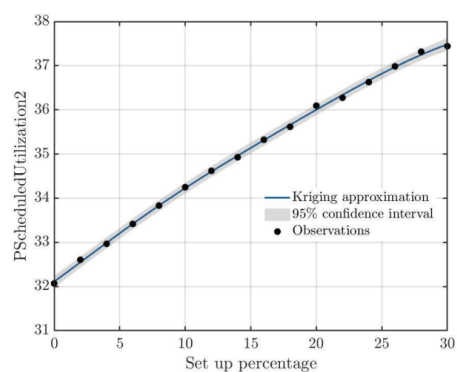
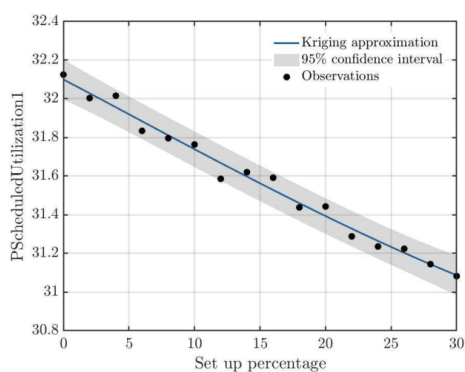




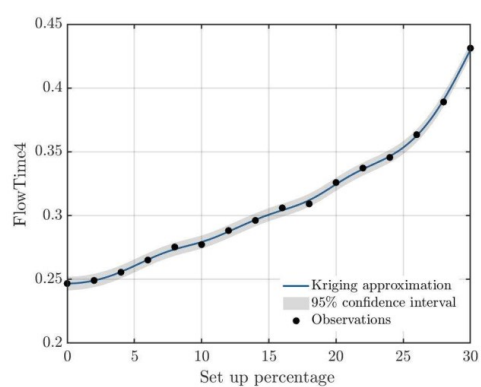
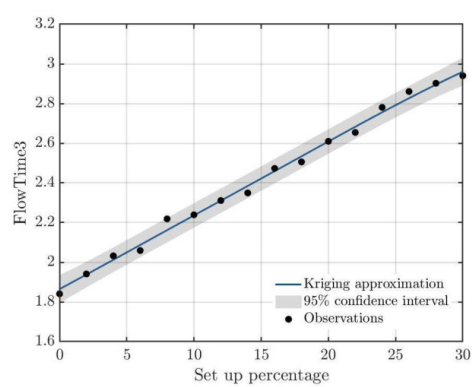
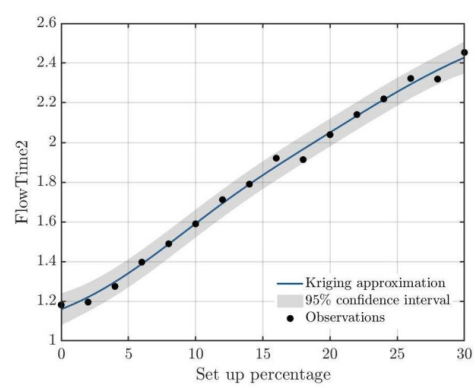
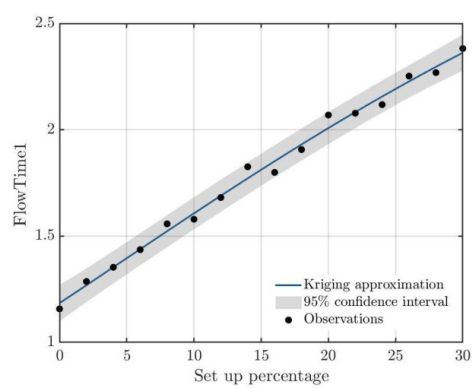
REGRA EARLIEST DUE DATE COM 10 MINUTOS NA POOL

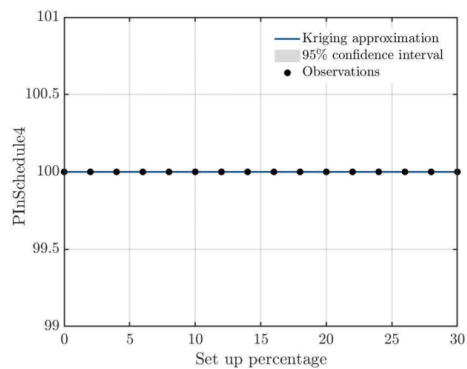
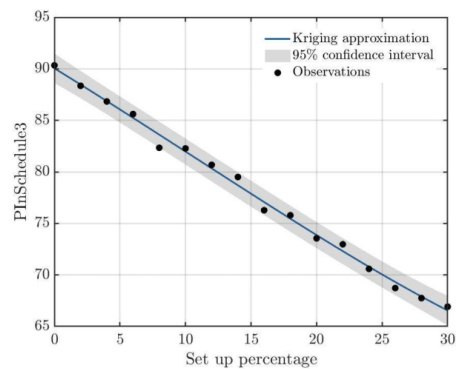
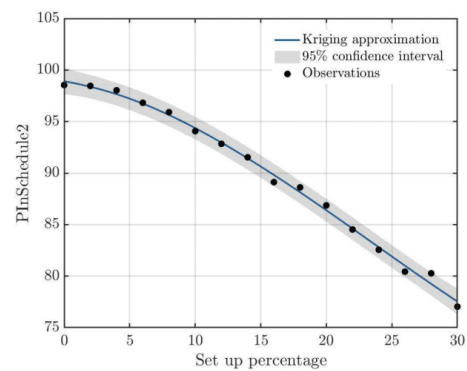
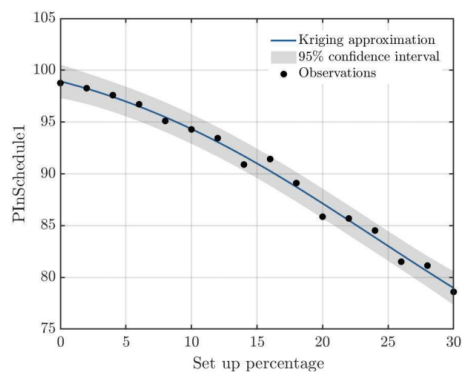
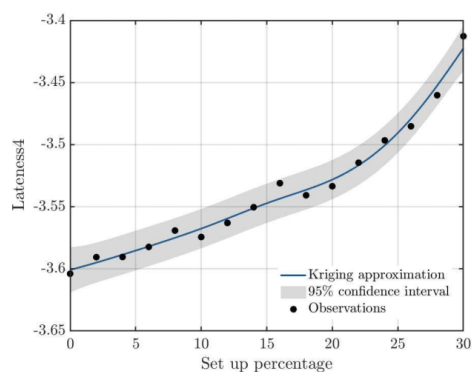
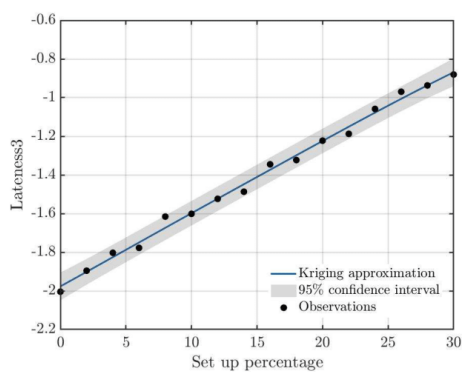
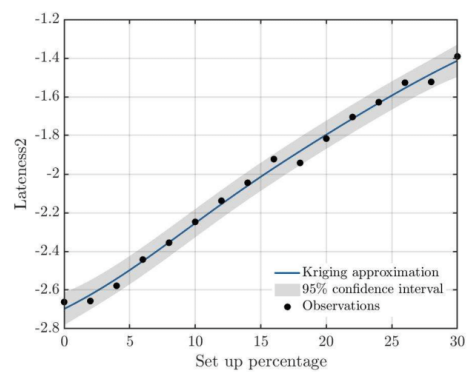
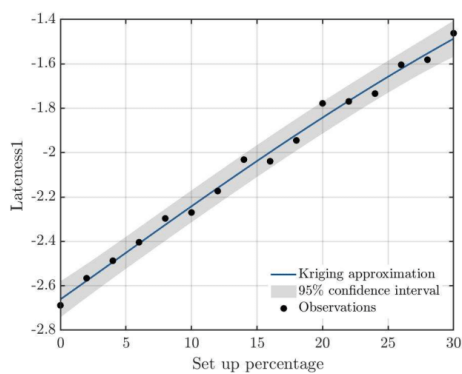


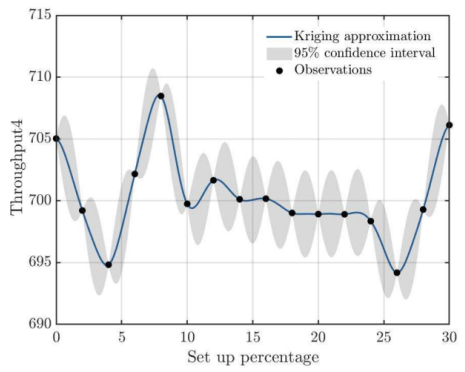
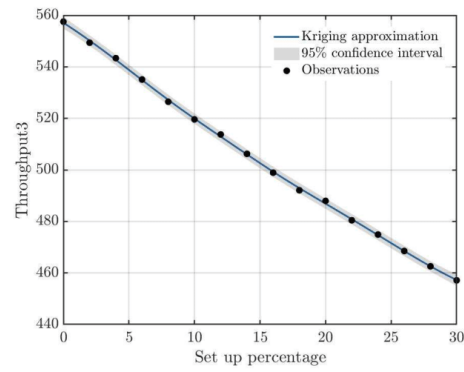
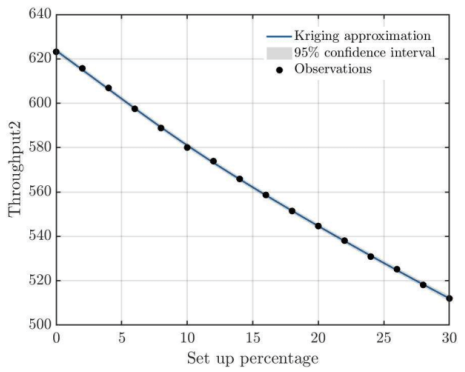
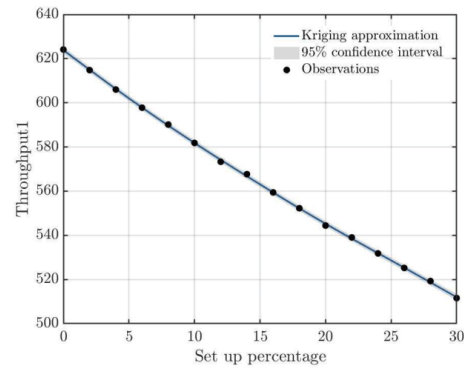
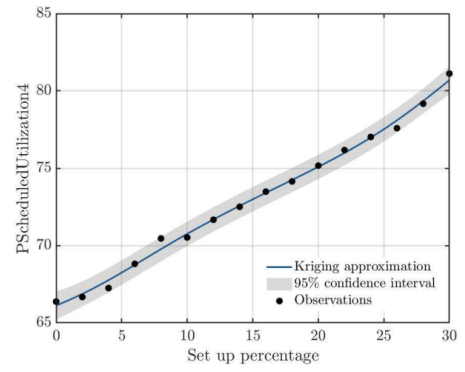
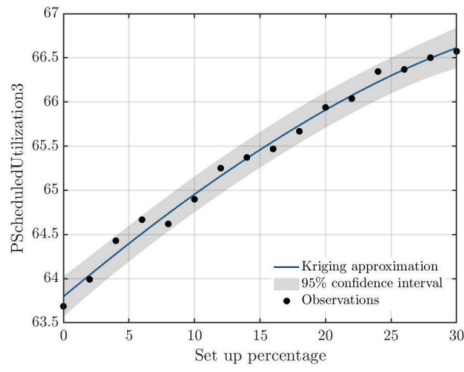
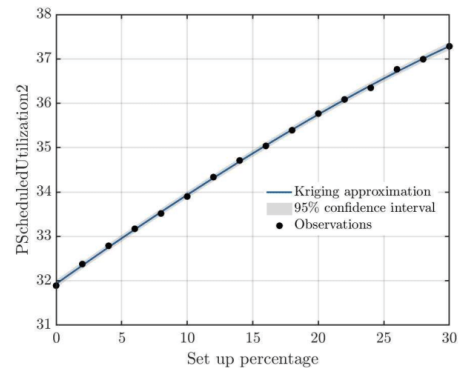
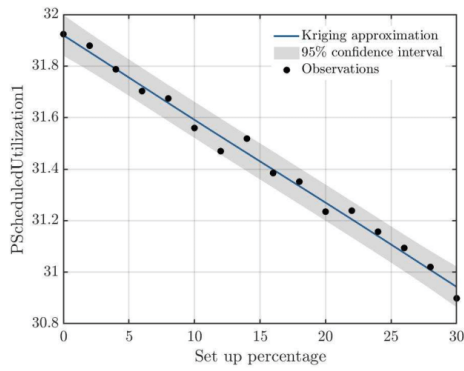




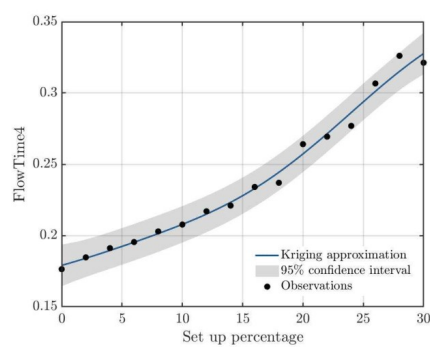
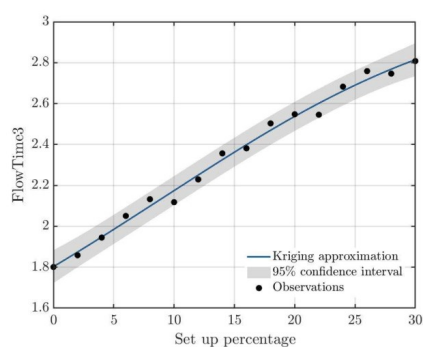
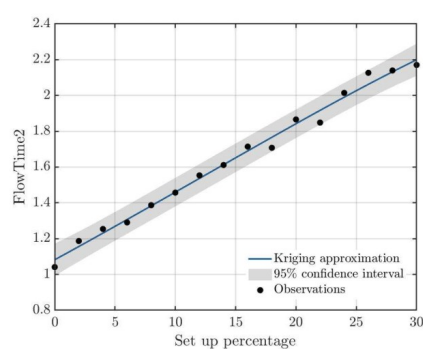
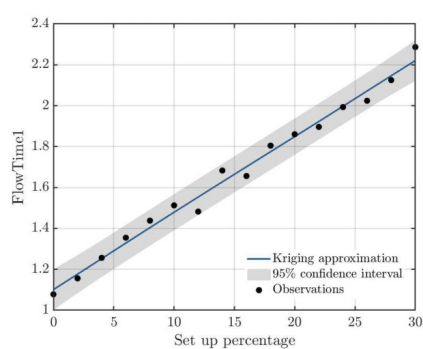
REGRA EARLIEST DUE DATE COM 15 MINUTOS NA POOL

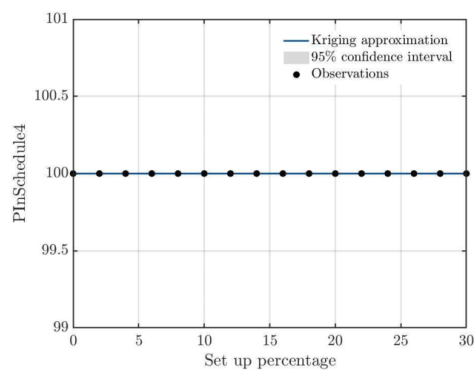
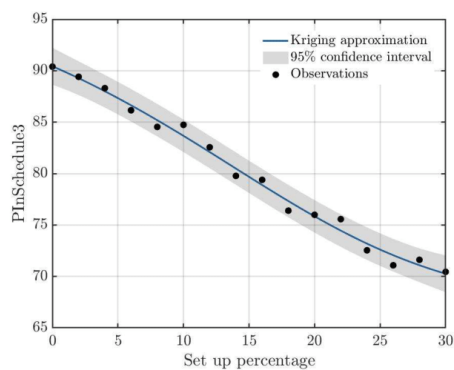
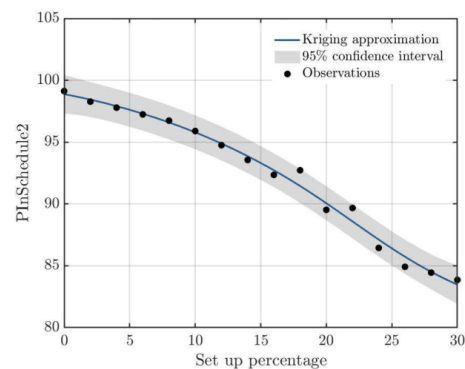
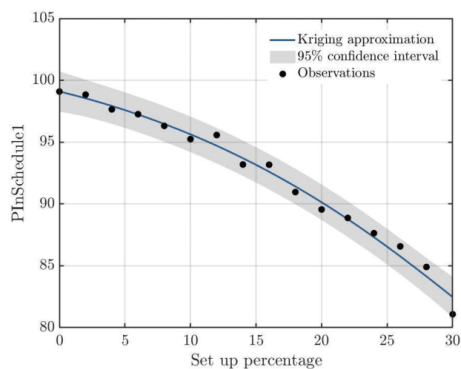
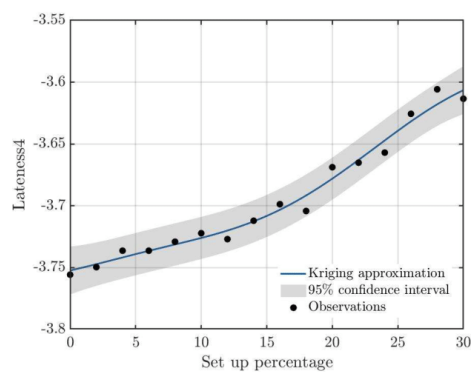
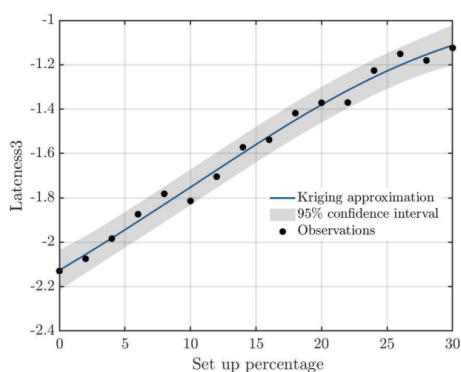
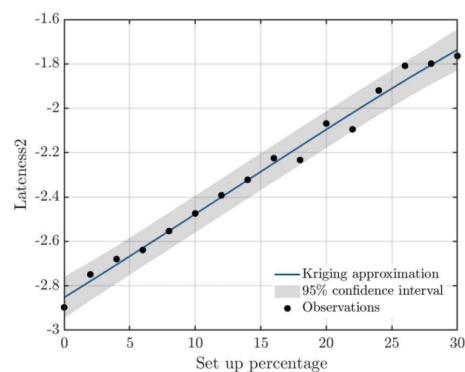
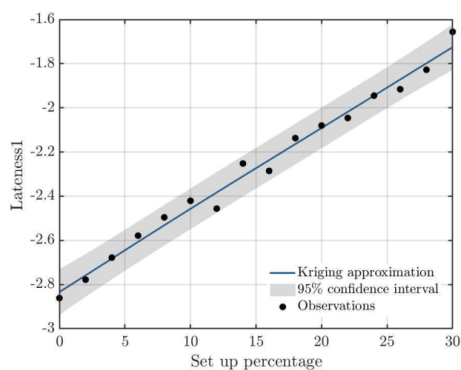


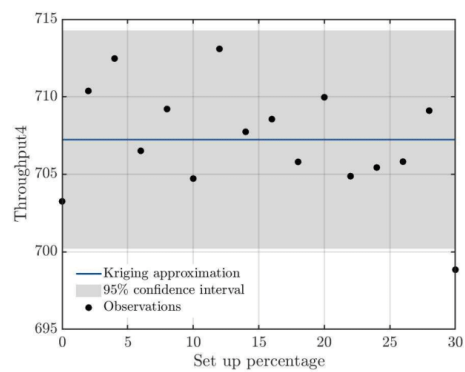
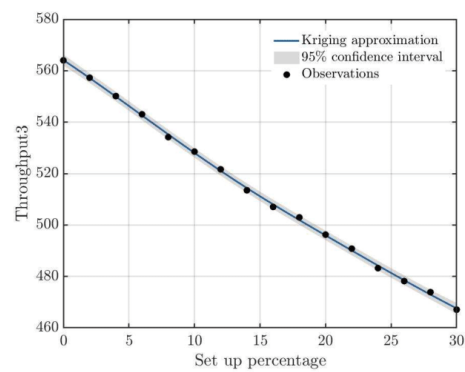
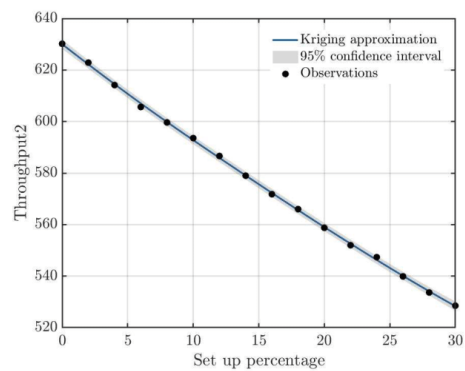
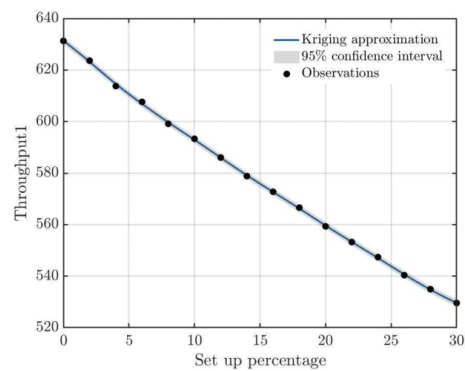
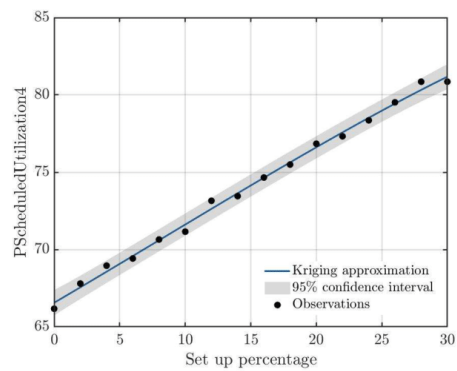
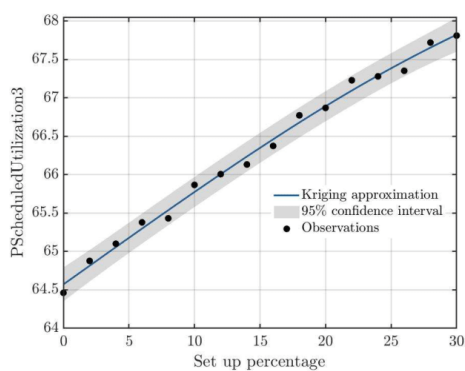
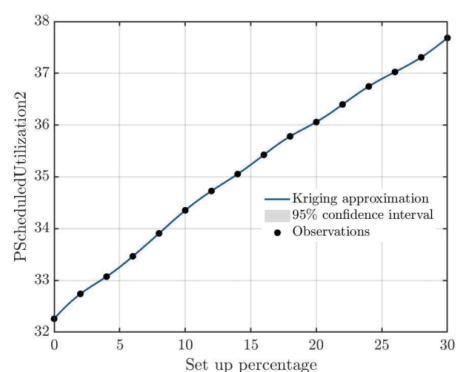
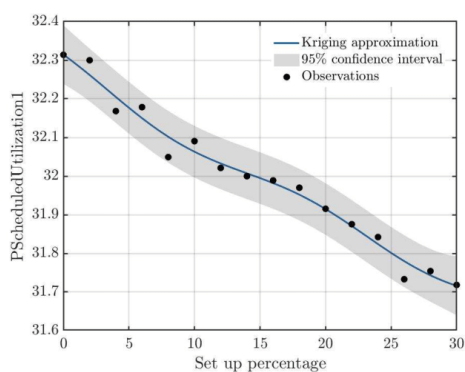




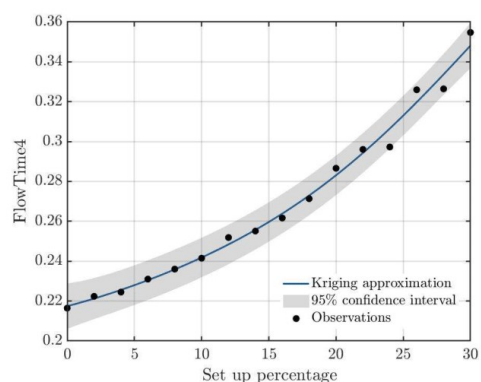
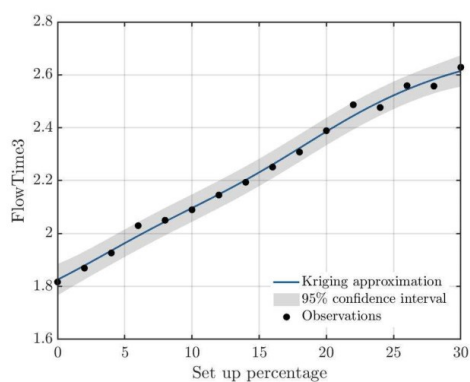
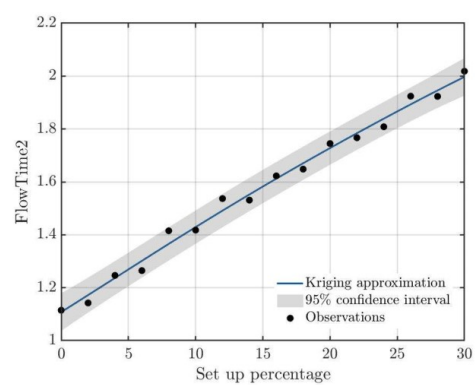
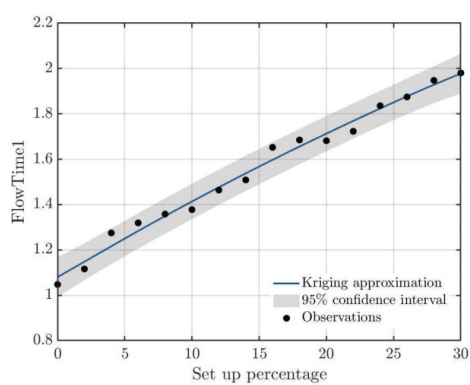
REGRA SHORTEST TOTAL PROCESSING TIME COM 5 MINUTOS NA POOL

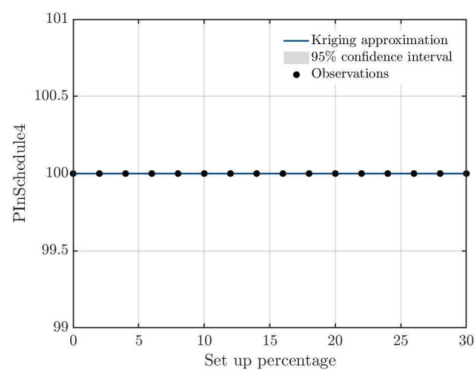
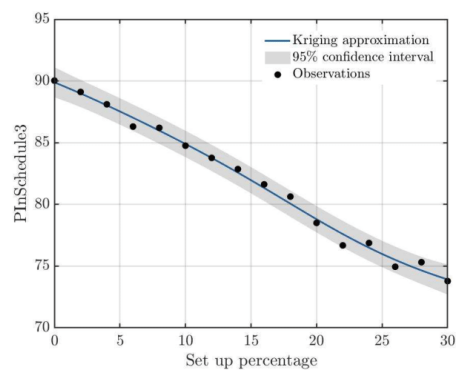
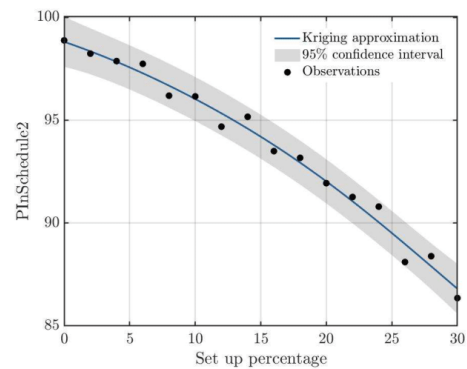
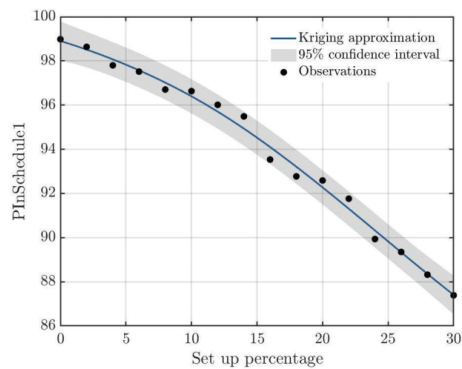
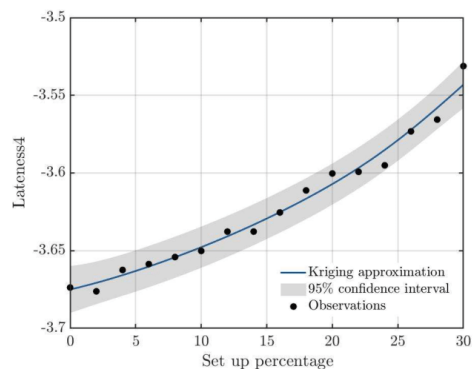
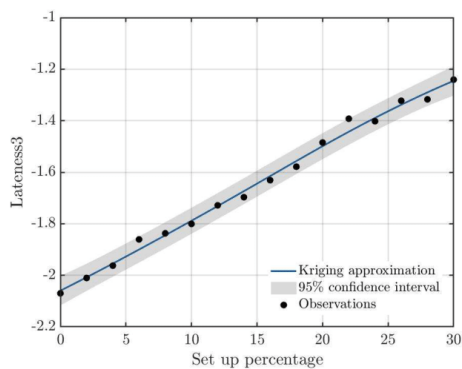
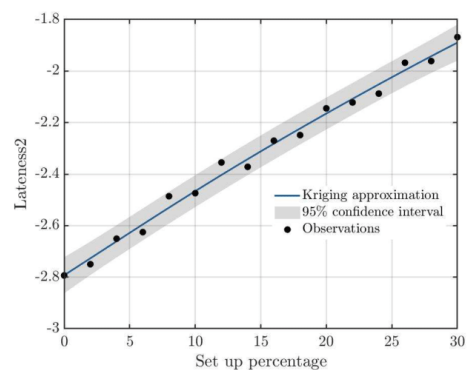
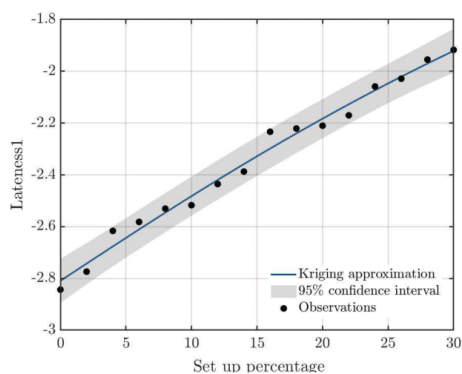


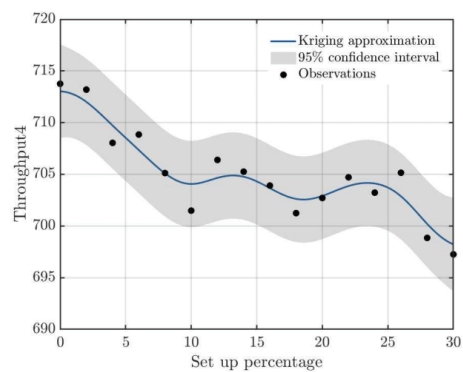
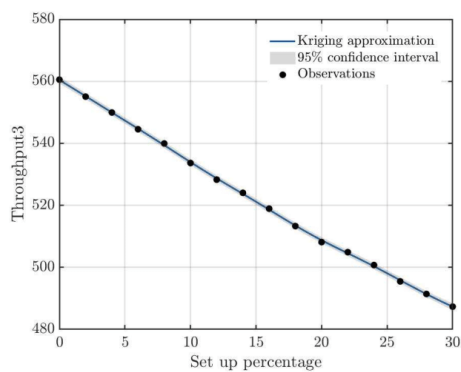
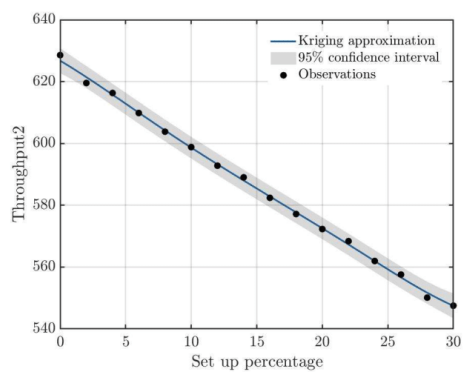
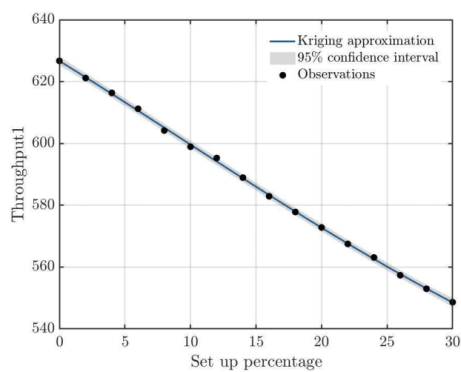
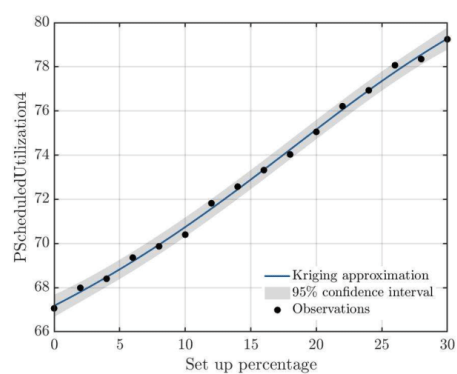
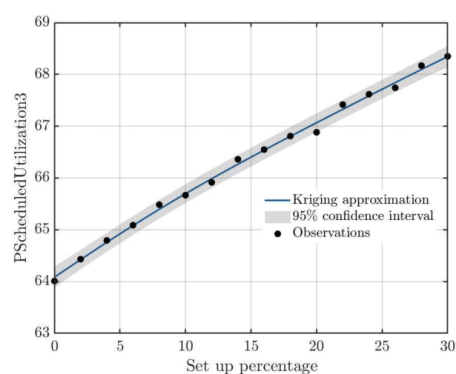
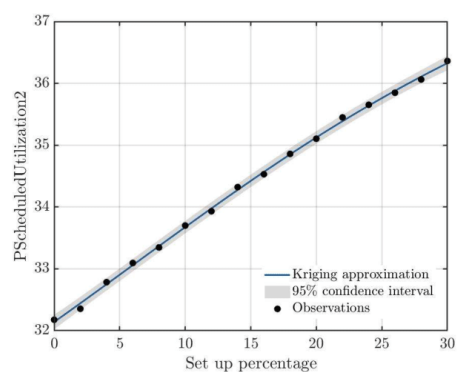
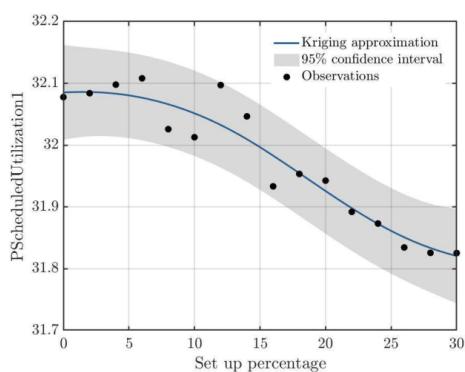




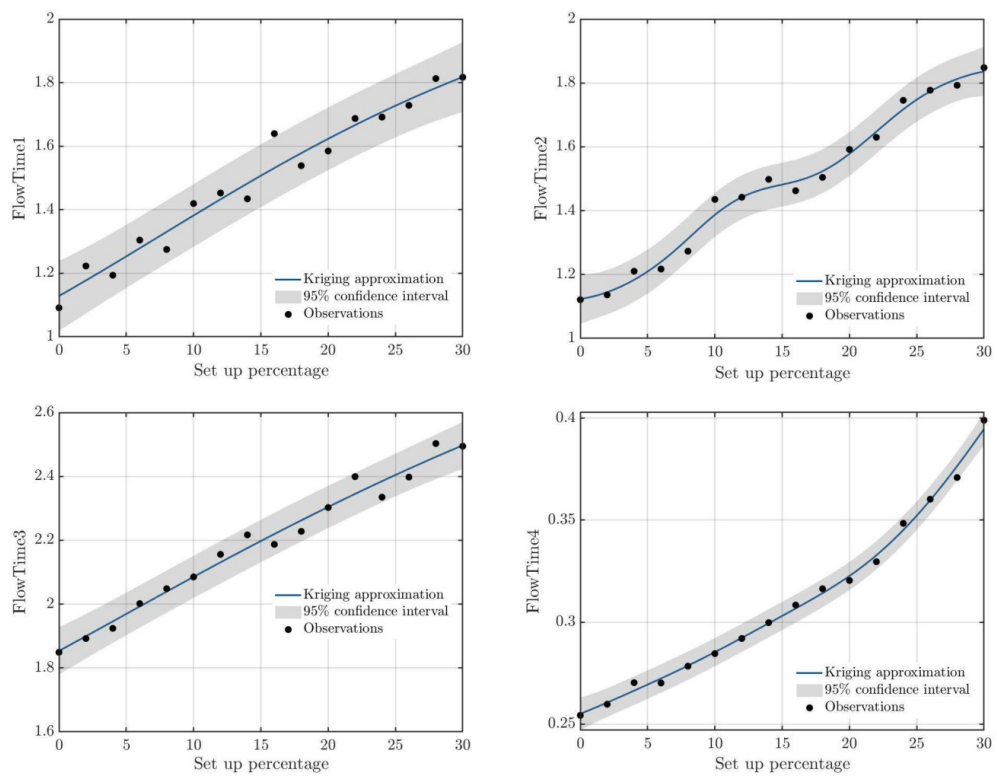
REGRA SHORTEST TOTAL PROCESSING TIME COM 10 MINUTOS NA POOL

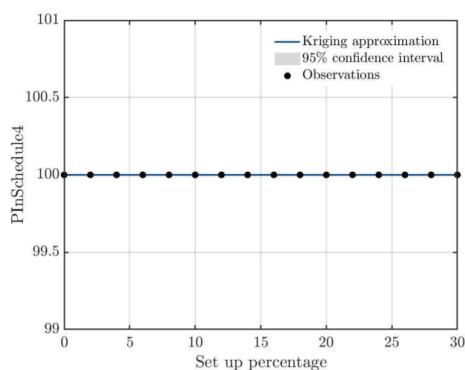
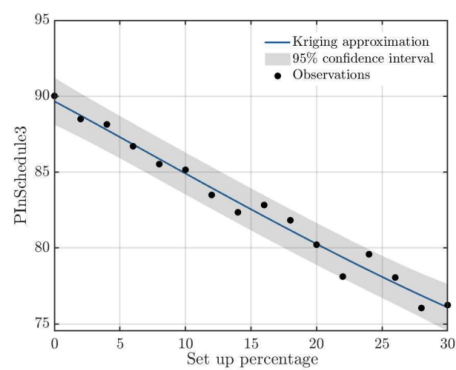
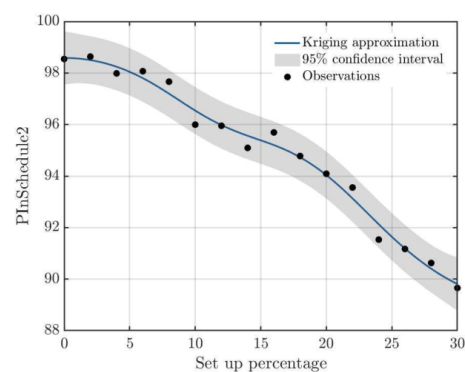
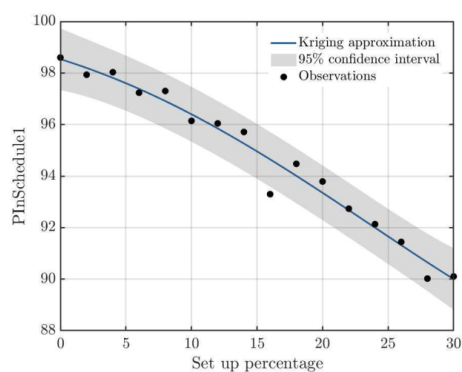
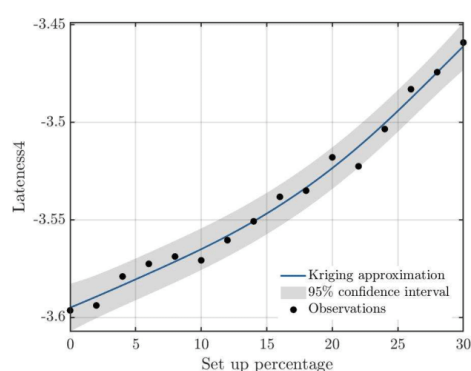
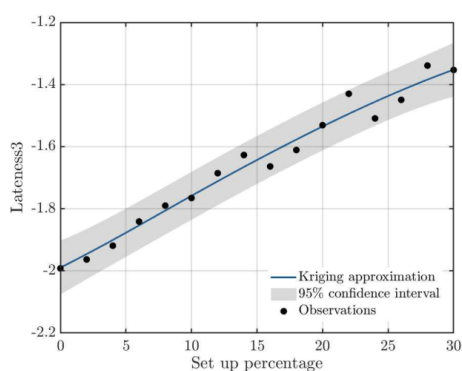
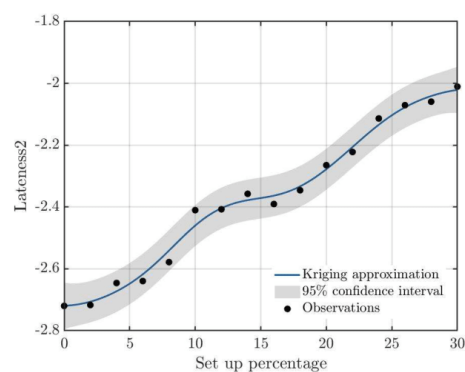
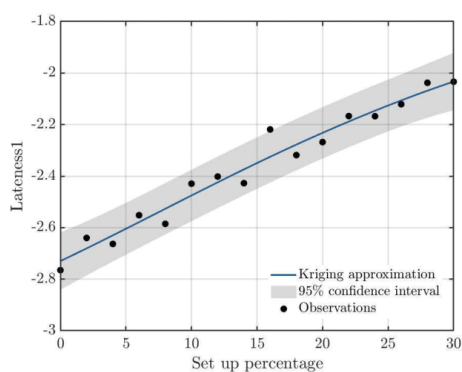


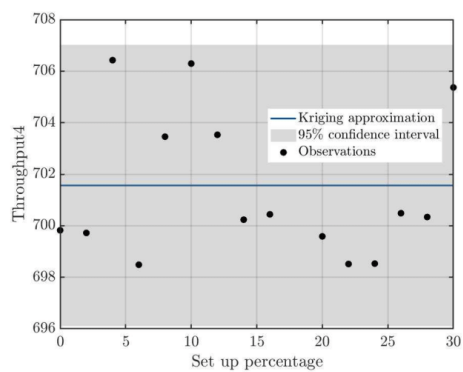
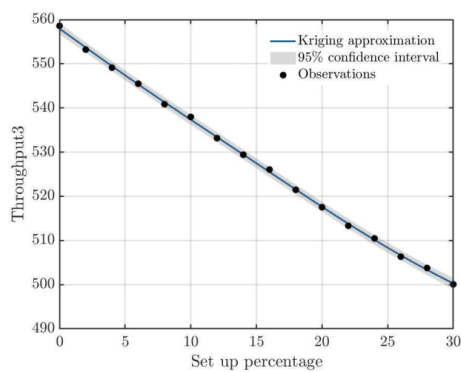
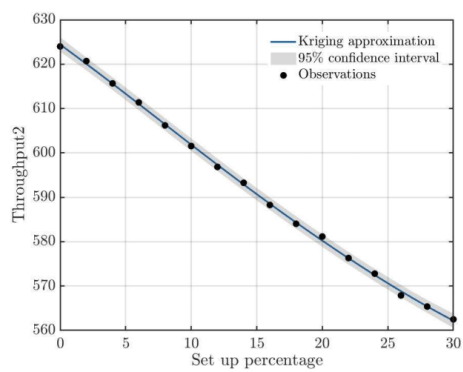
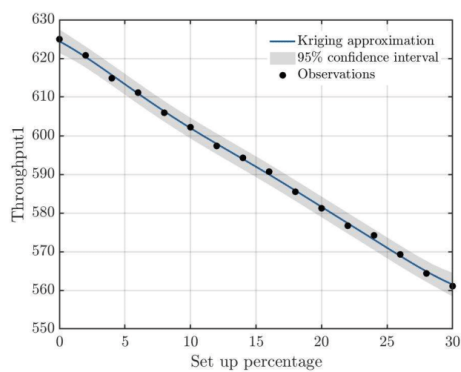
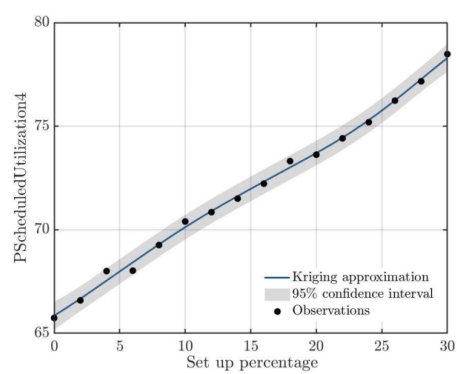
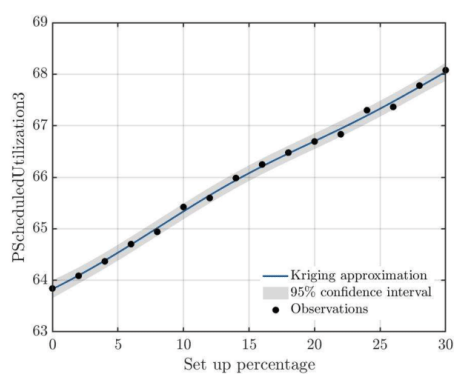
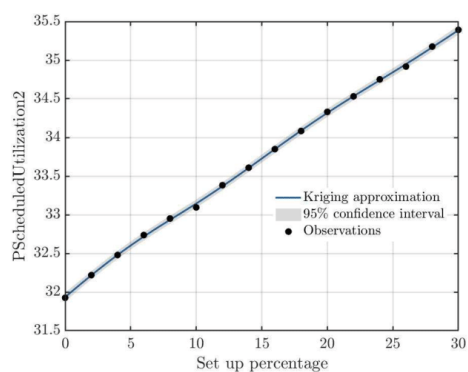
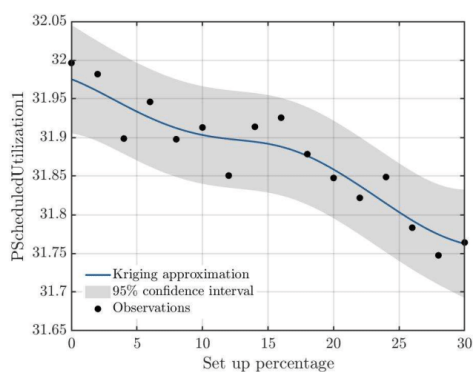




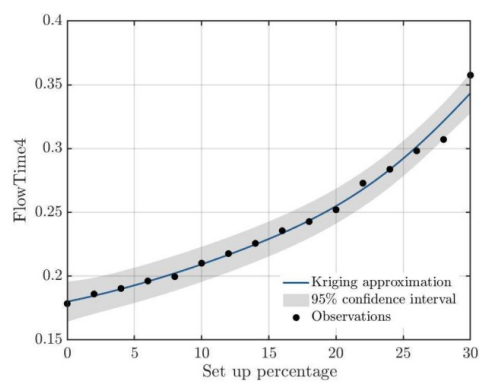
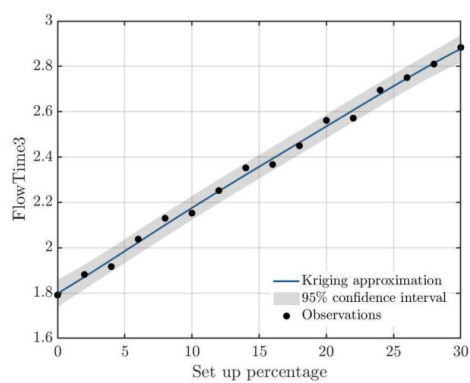
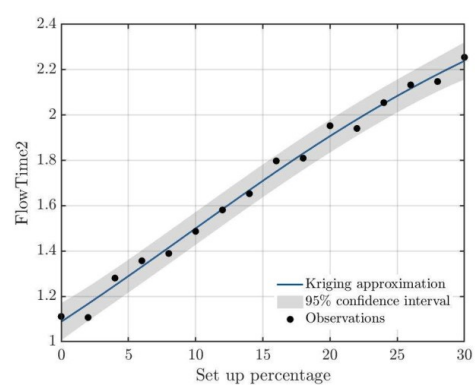
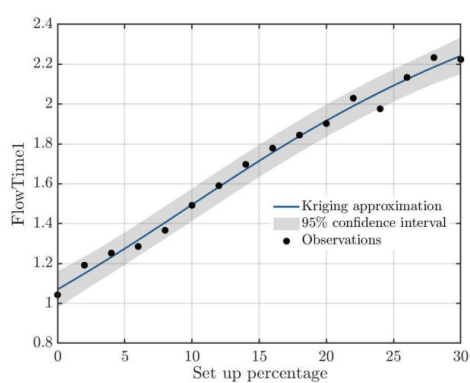
REGRA SHORTEST TOTAL PROCESSING TIME COM 15 MINUTOS NA POOL

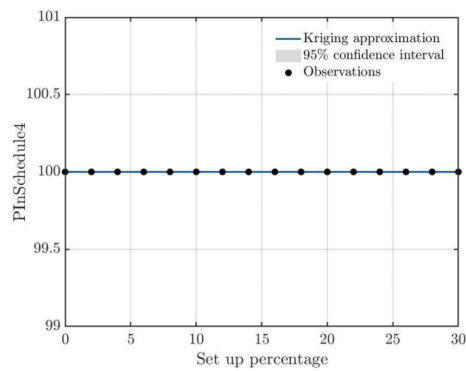
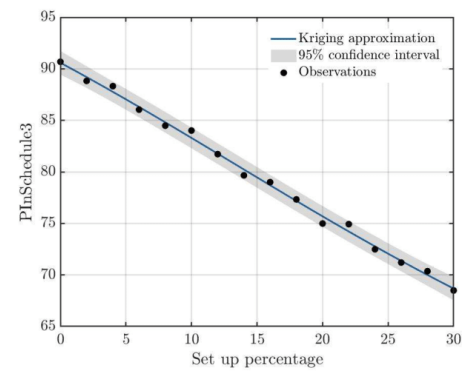
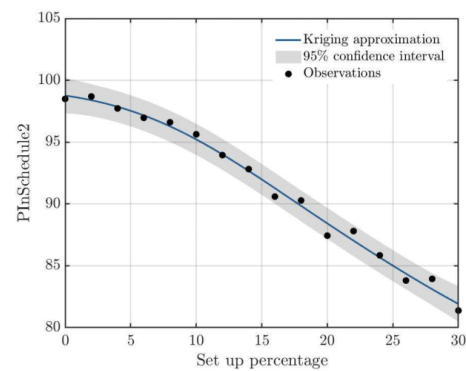
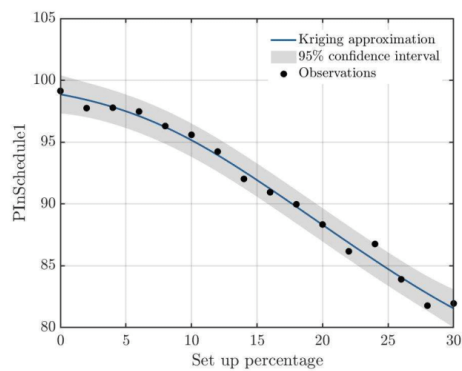
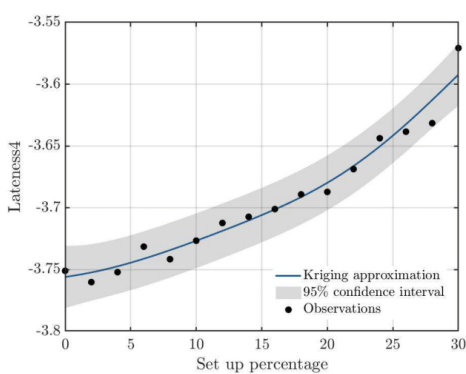
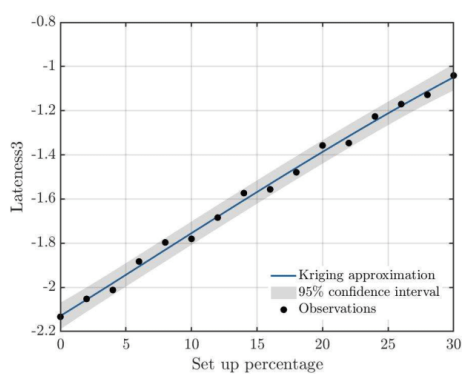
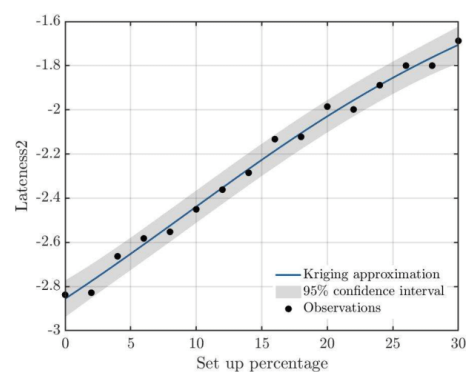
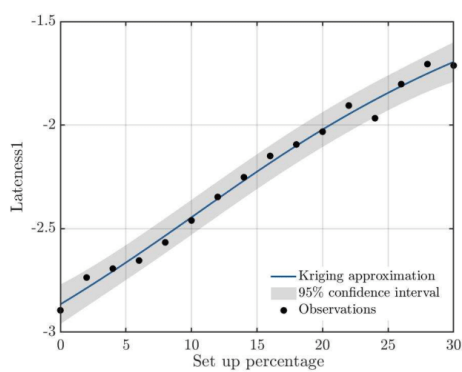


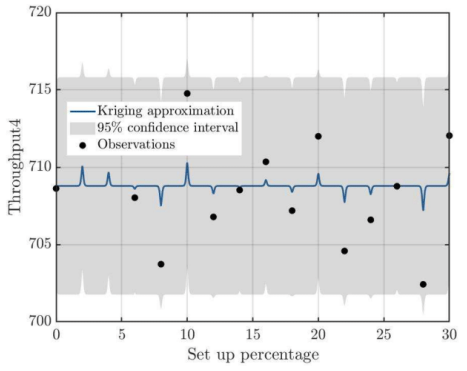
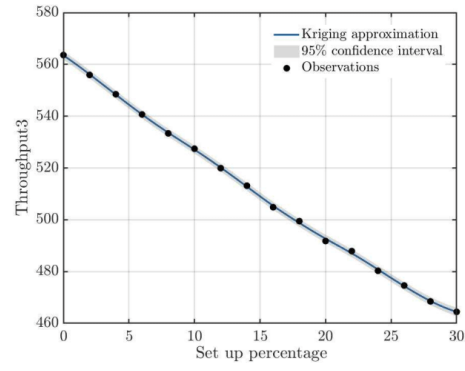
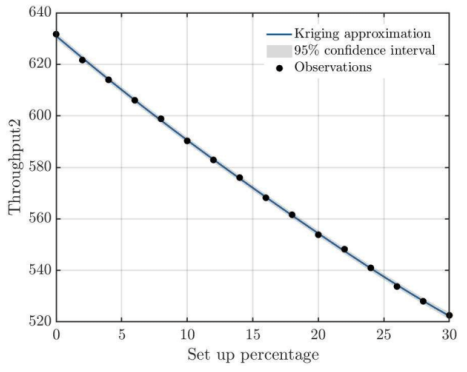
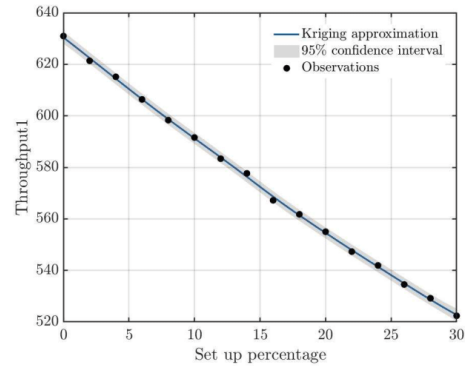
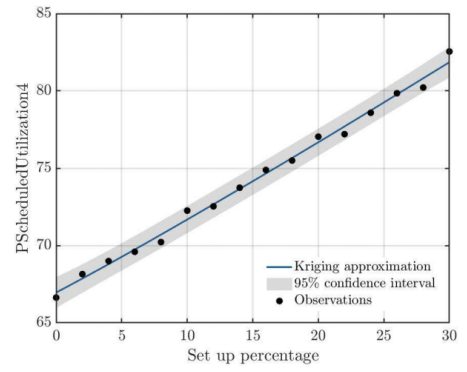
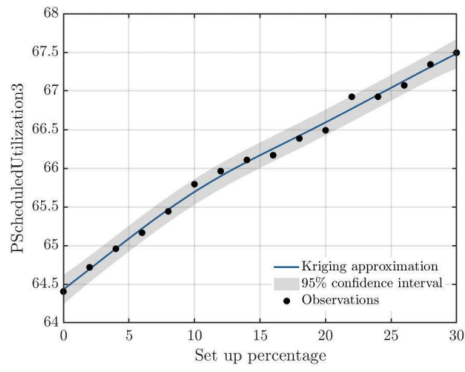
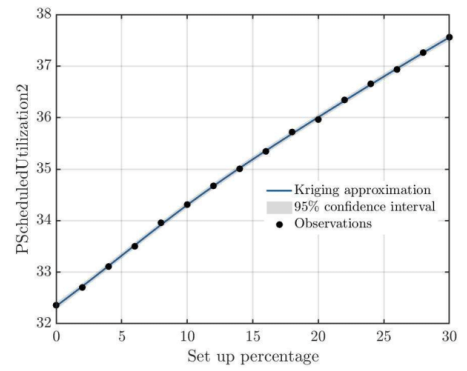
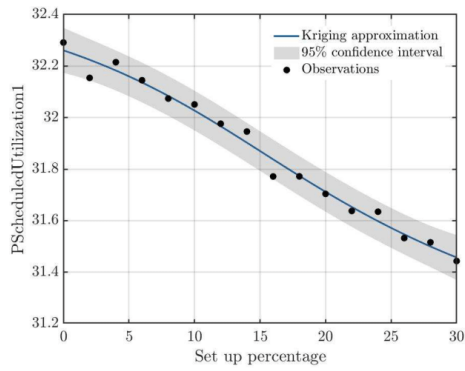




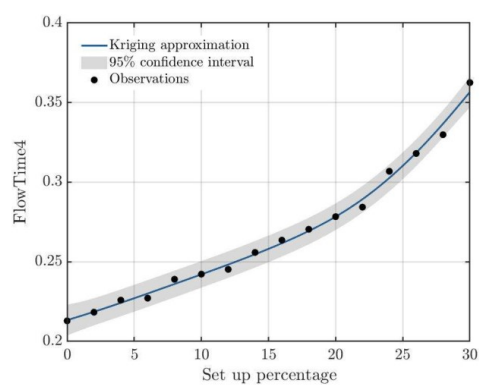
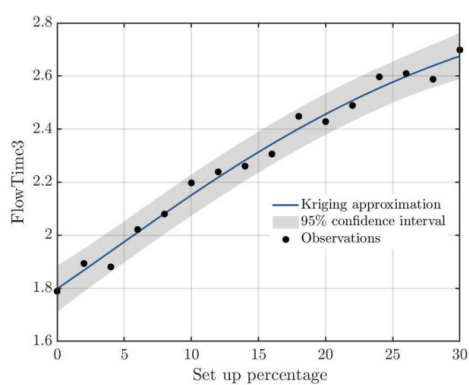
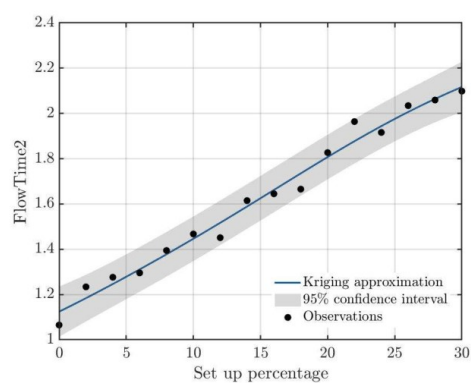
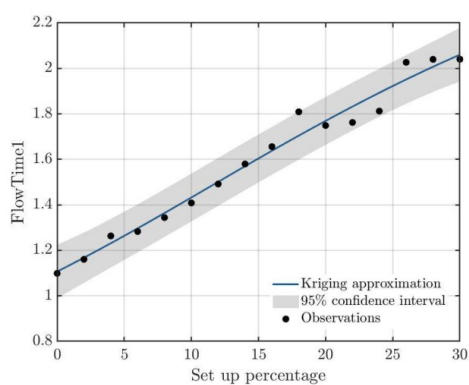
REGRA LONGEST PROCESS SEQUENCE COM 5 MINUTOS NA POOL

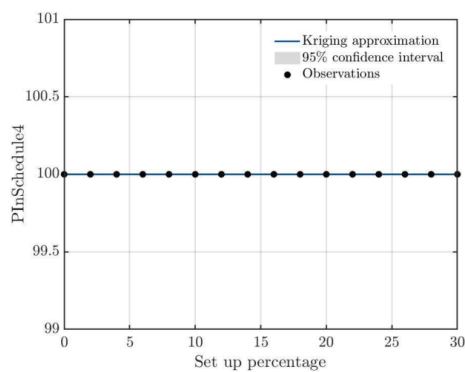
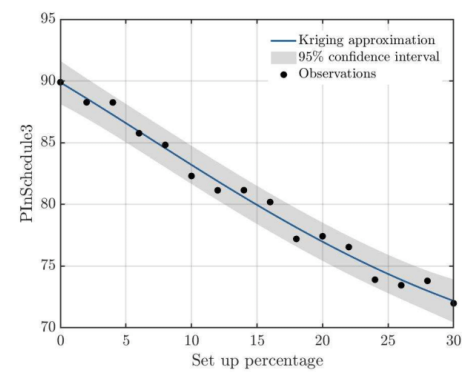
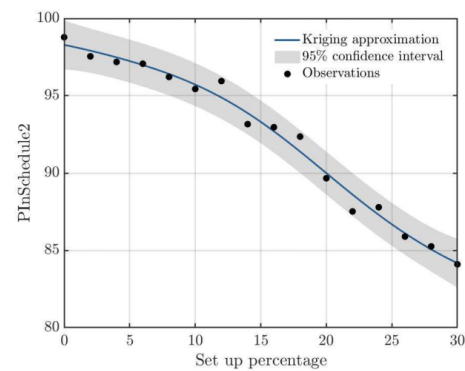
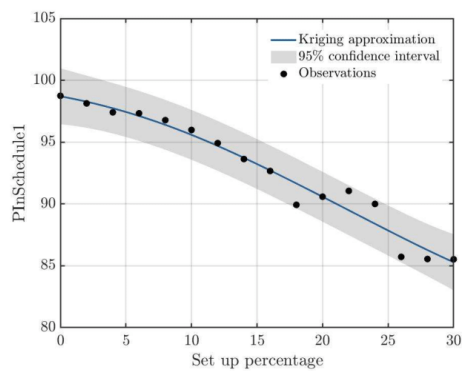
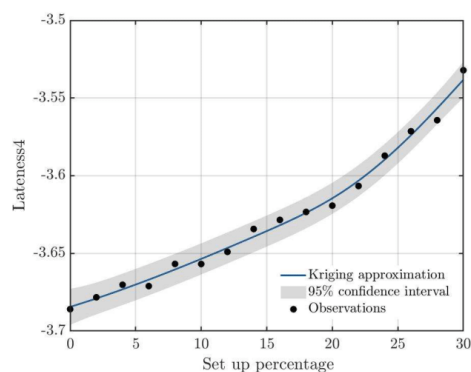
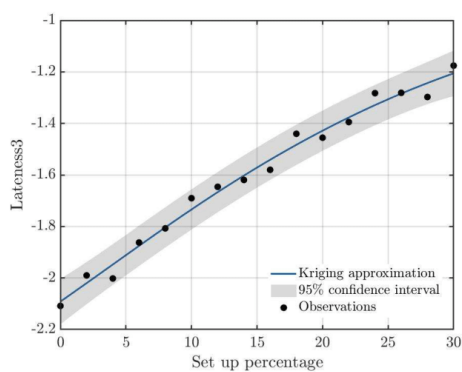
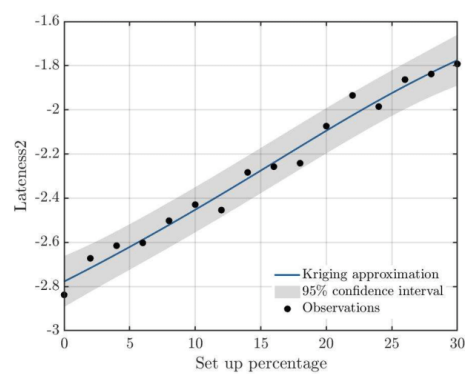
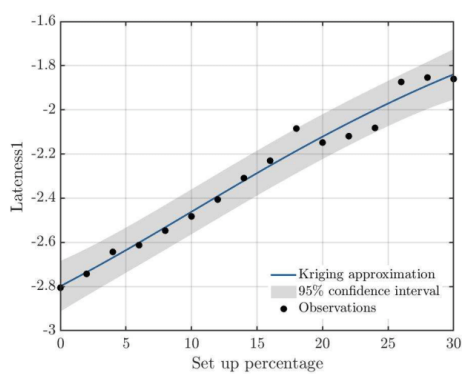


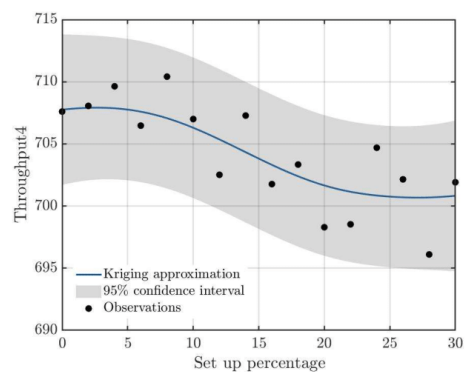
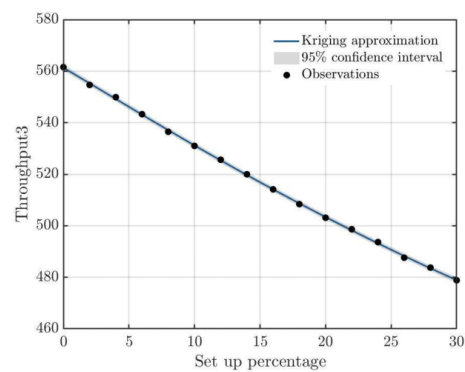
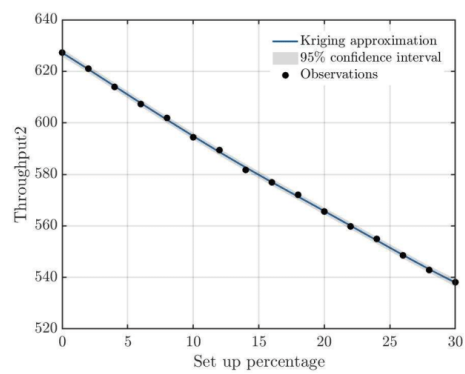
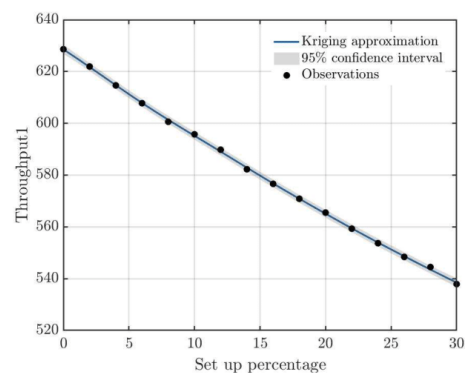
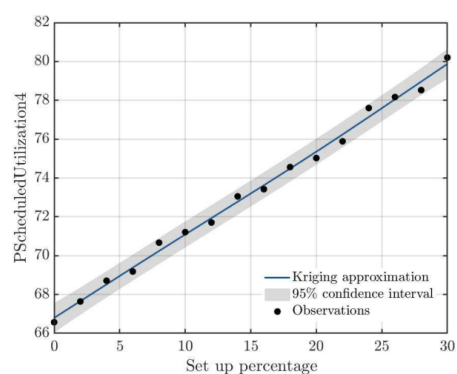
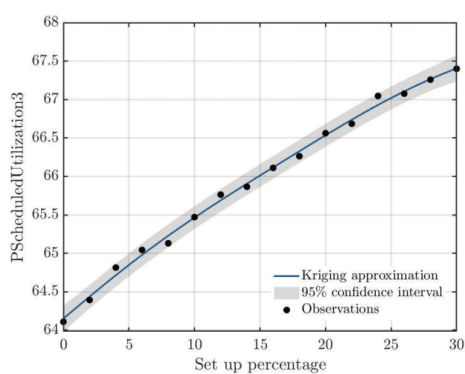
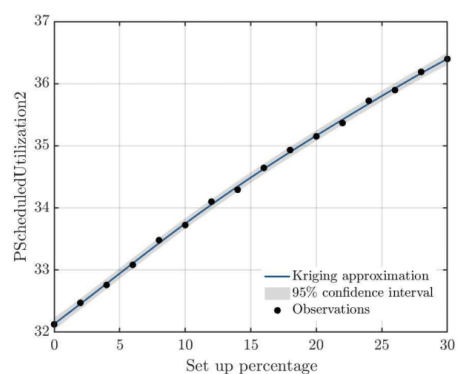
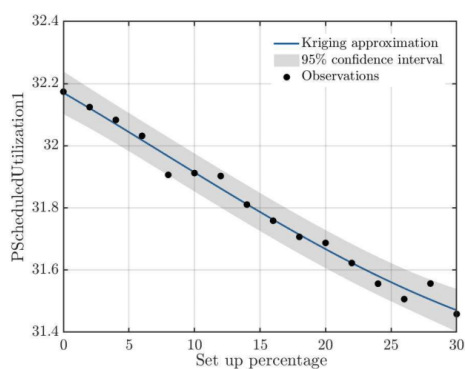




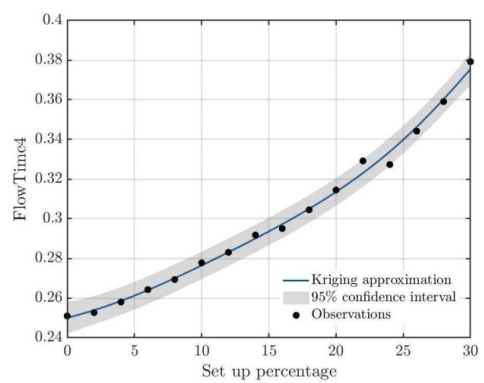
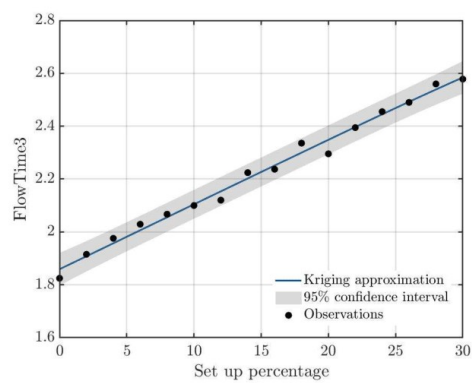
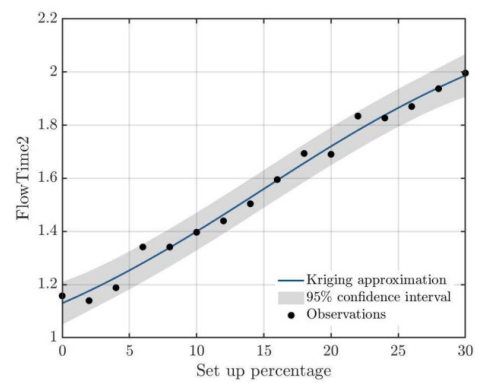
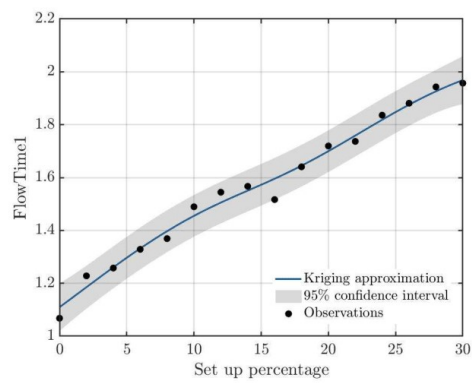
REGRA LONGEST PROCESS SEQUENCE COM 10 MINUTOS NA POOL

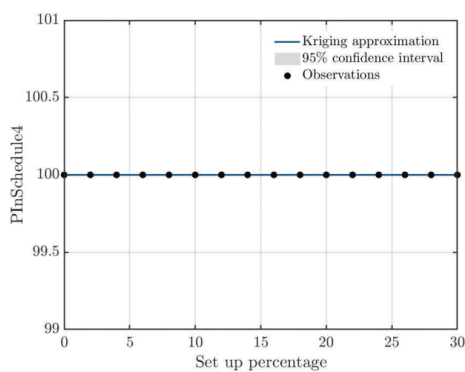
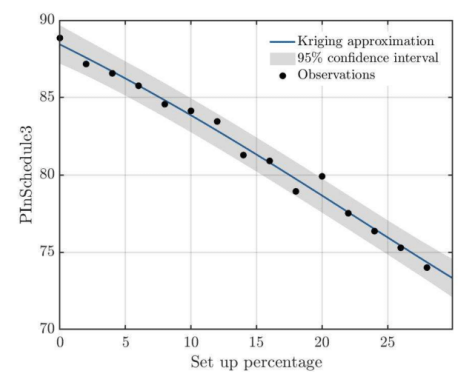
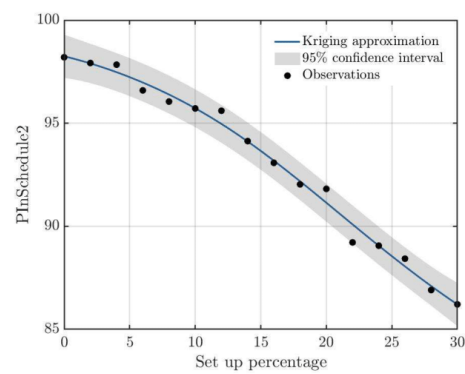
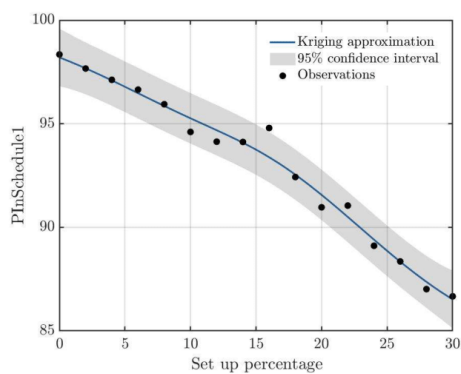
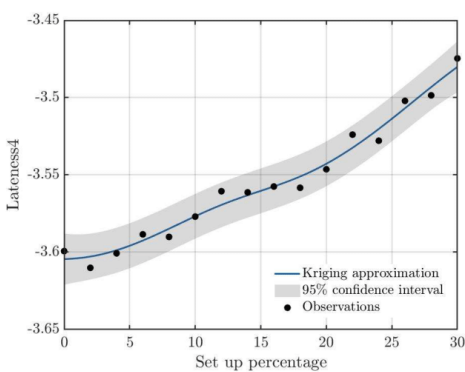
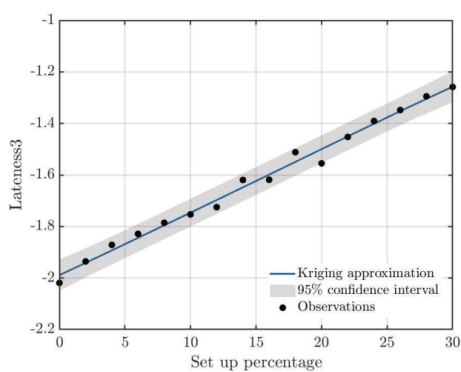
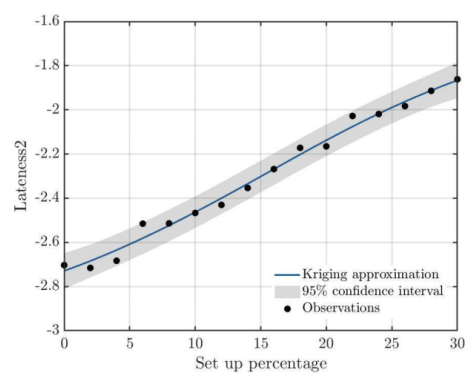
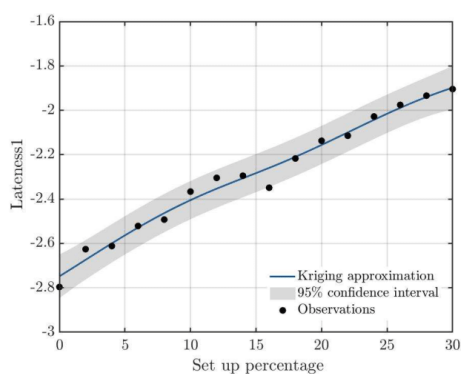


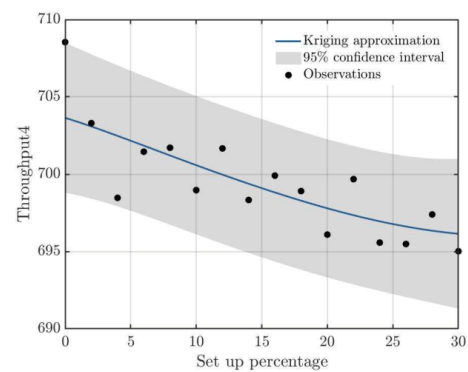
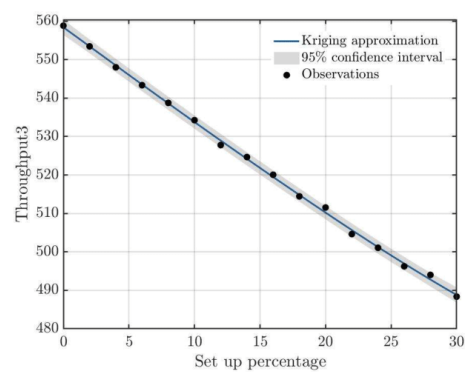
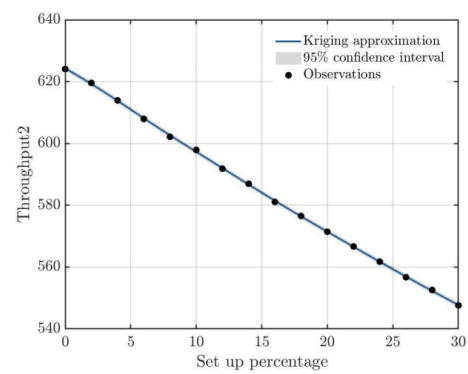
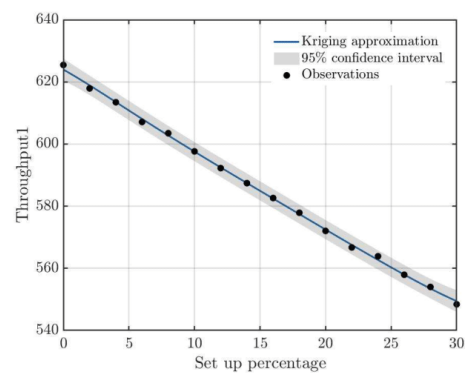
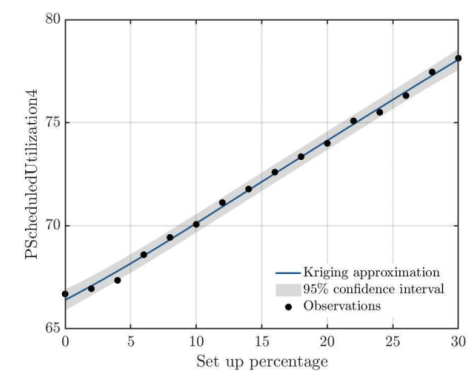
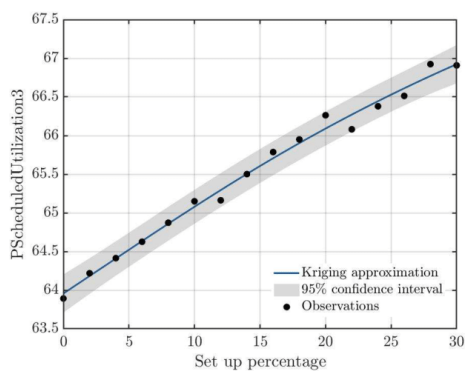
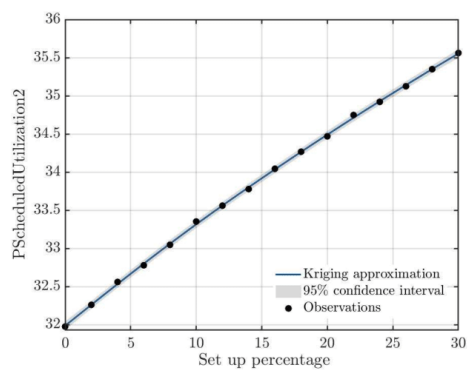
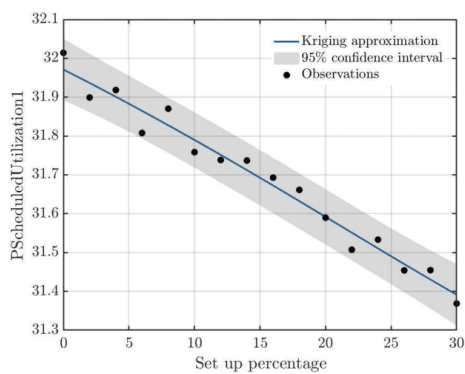




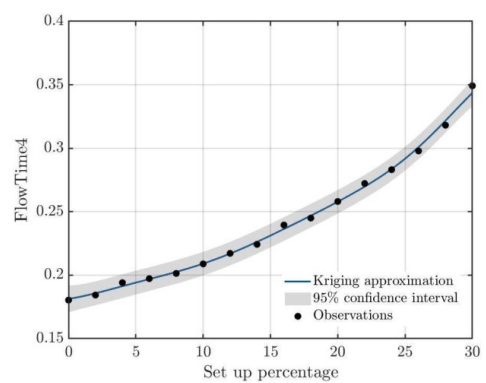
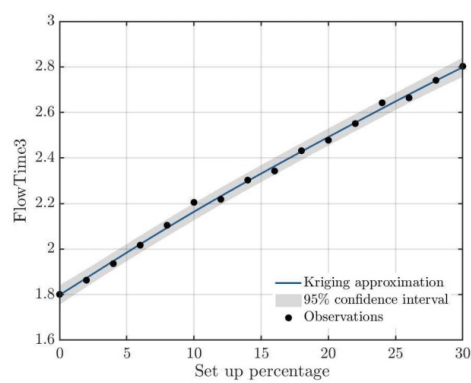
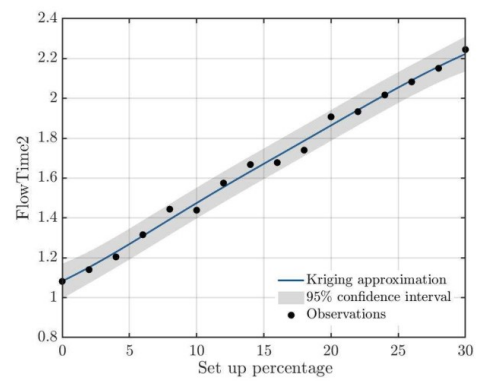
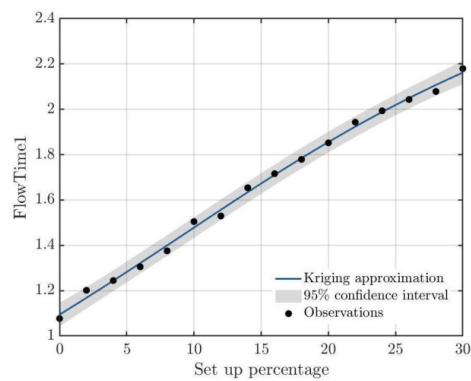
REGRA LONGEST PROCESS SEQUENCE COM 15 MINUTOS NA POOL

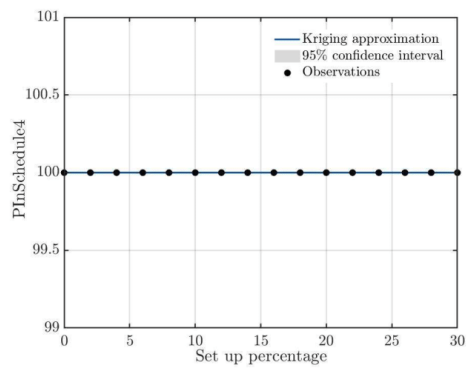
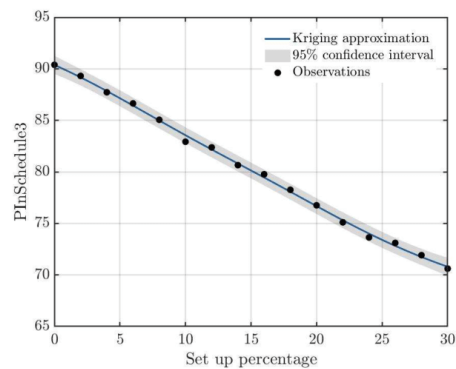
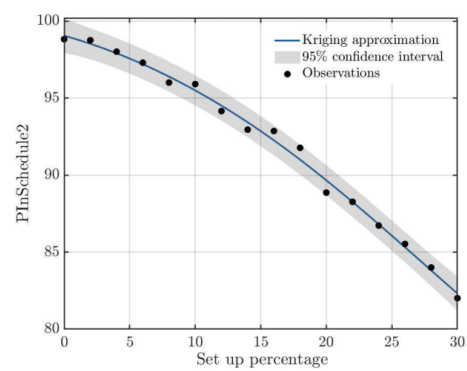
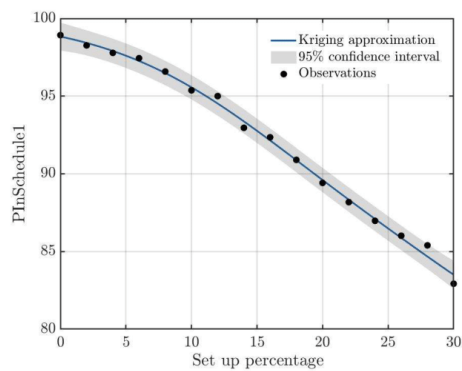
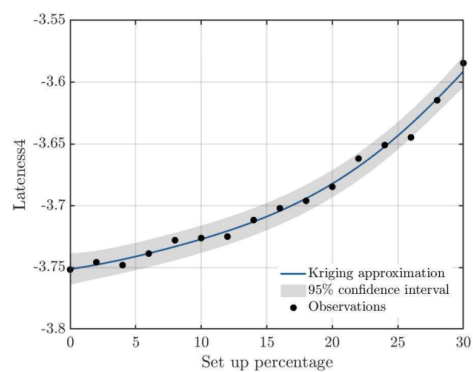
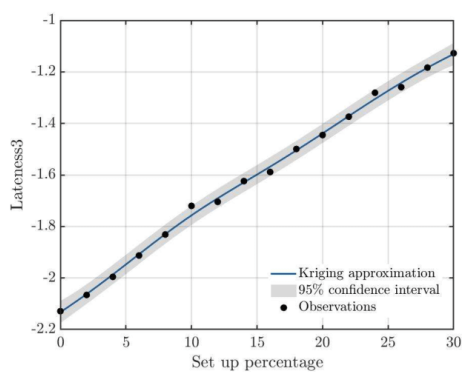
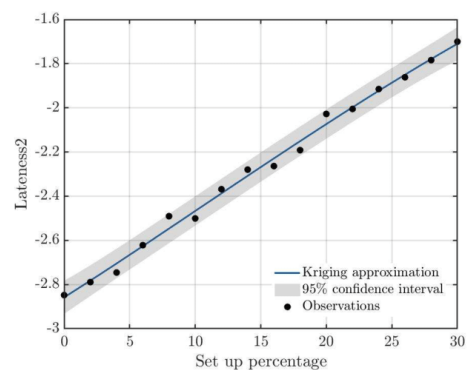
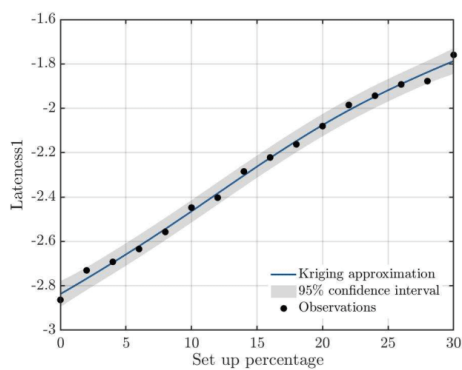


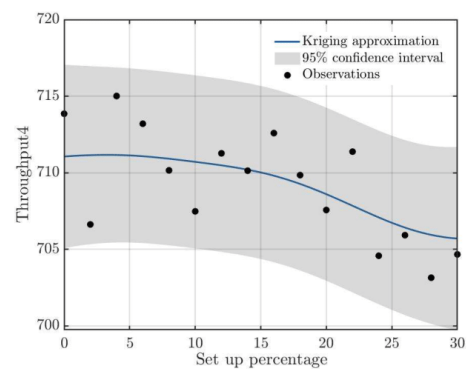
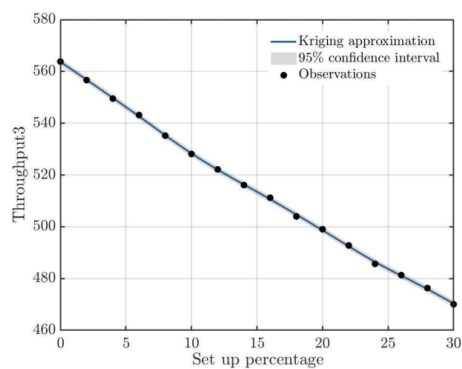
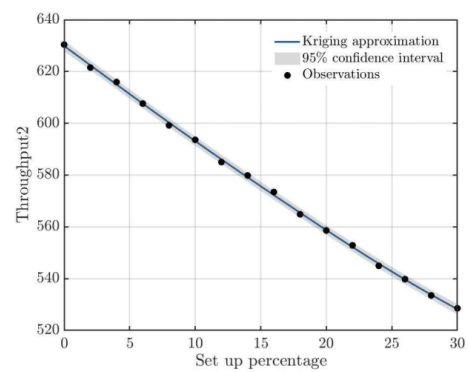
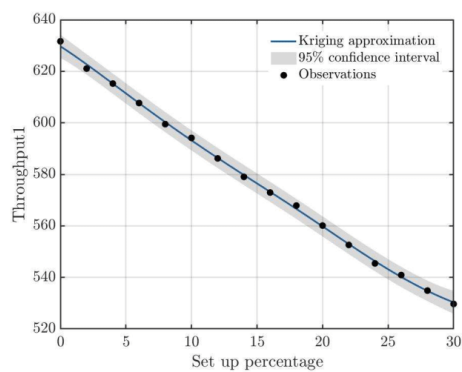
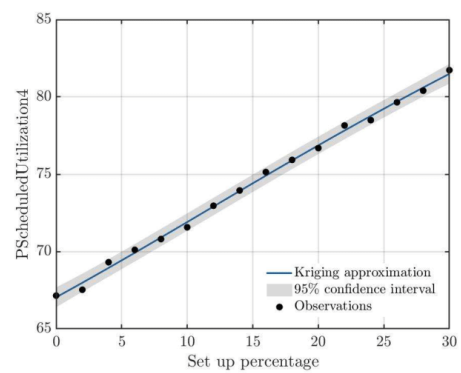
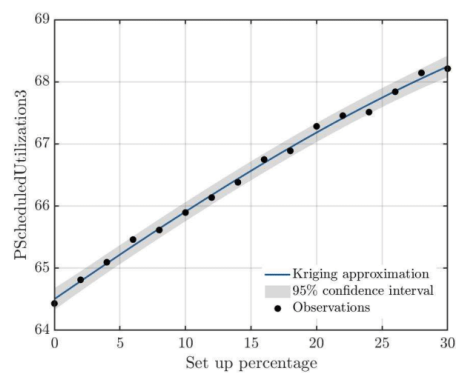
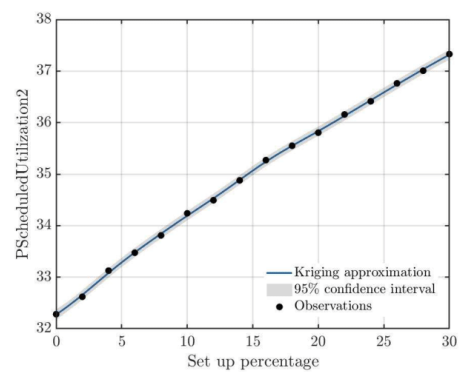
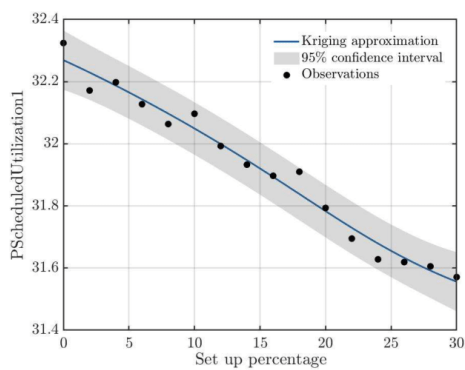




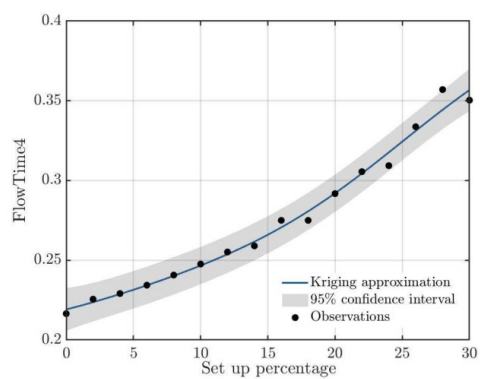
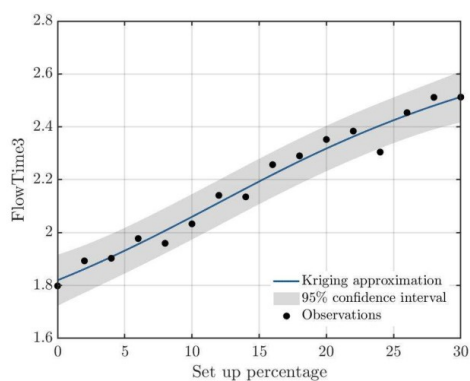
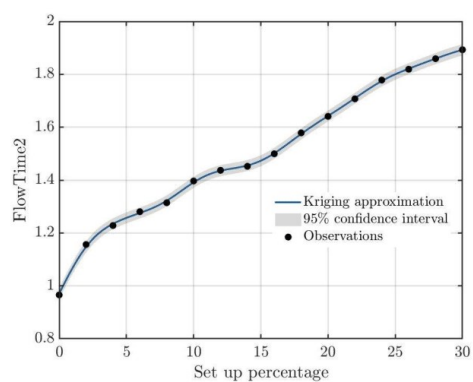
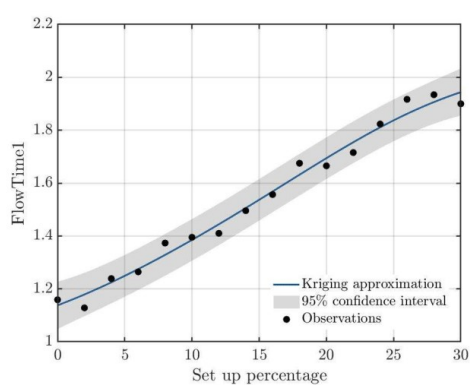
REGRA LONGEST PROCESSING TIME COM 5 MINUTOS NA POOL

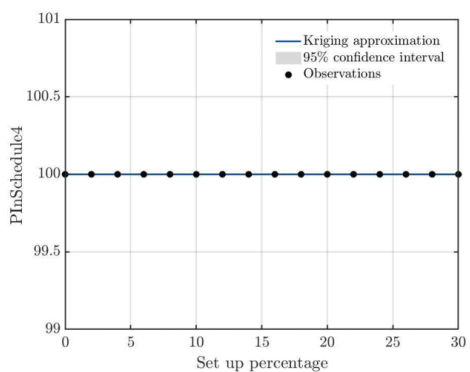
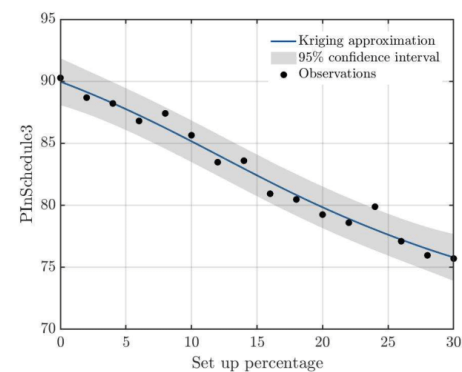
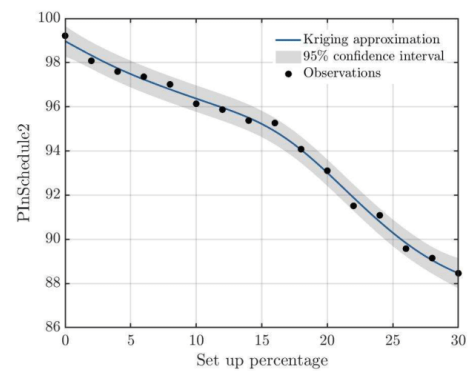
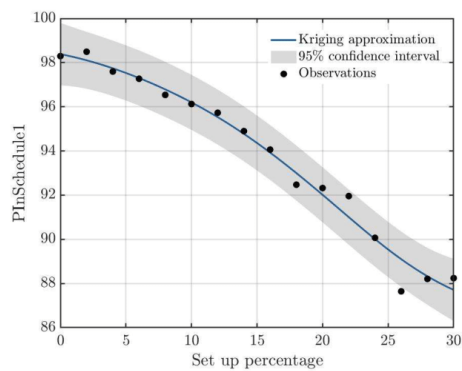
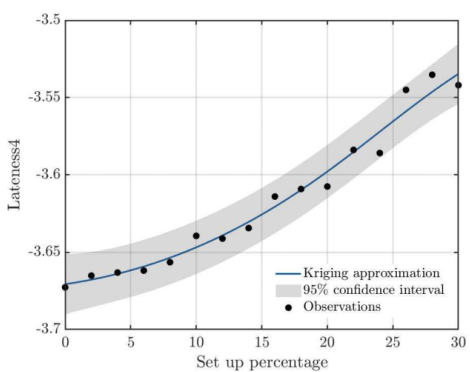
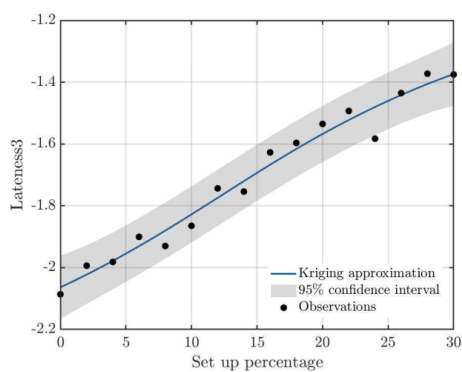
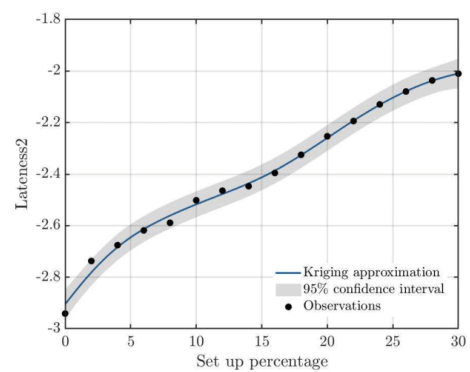
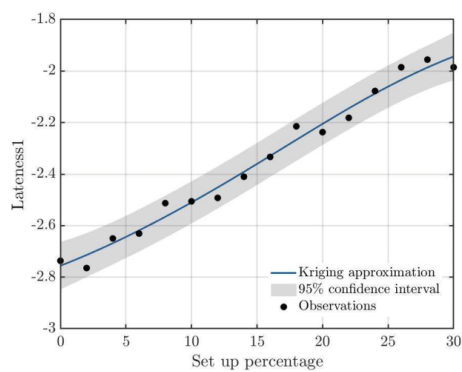


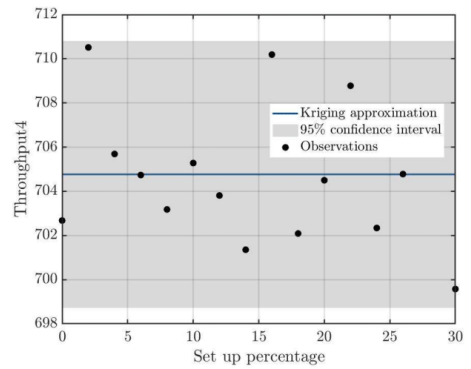
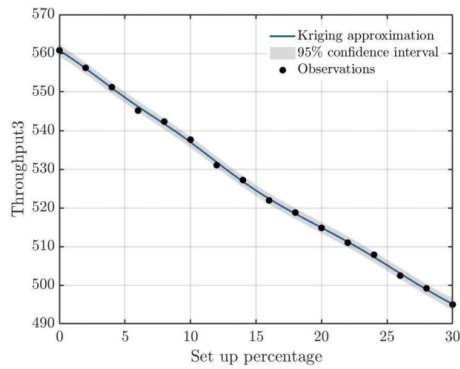
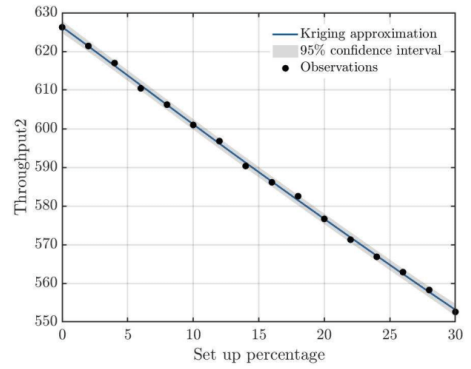
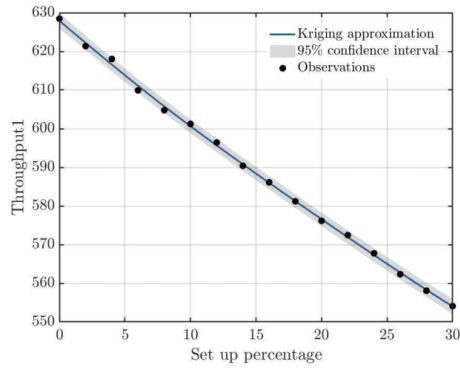
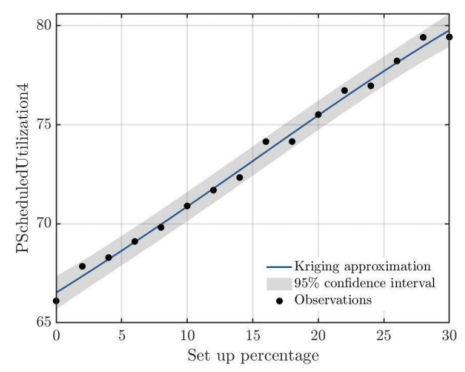
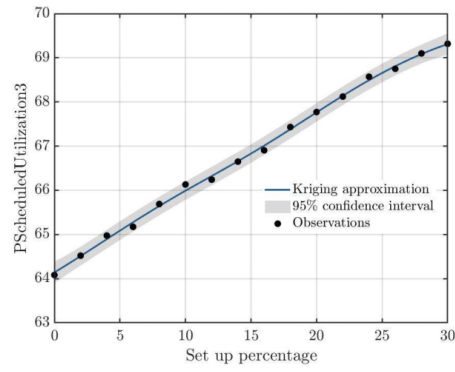
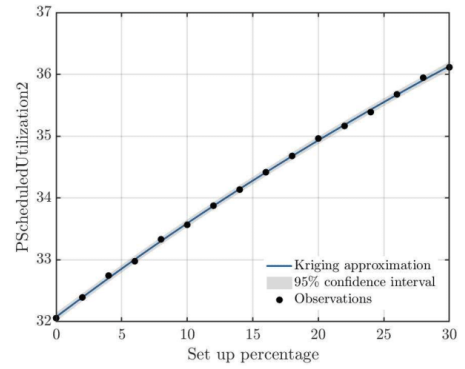
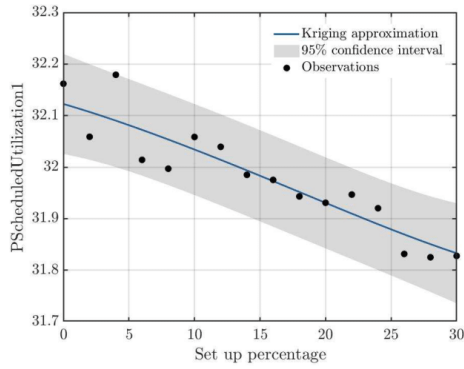




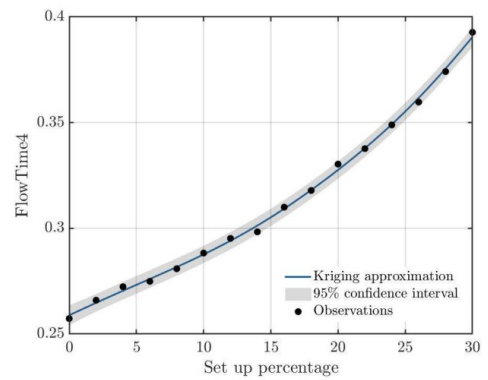
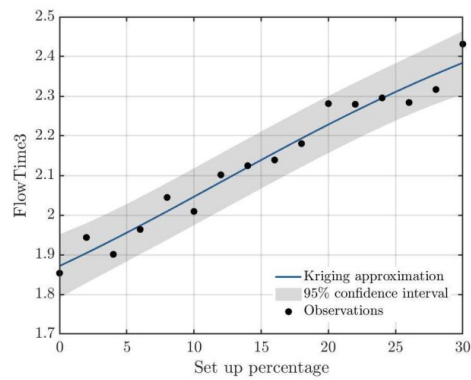
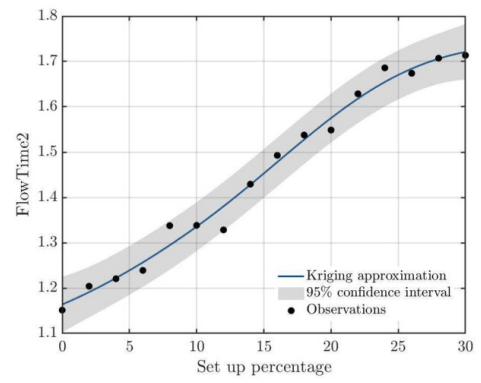
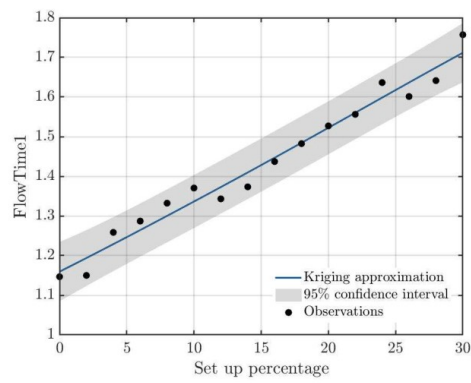
REGRA SHORTEST PROCESSING TIME COM 10 MINUTOS NA POOL

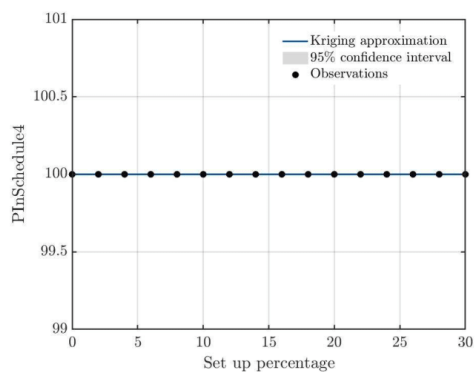
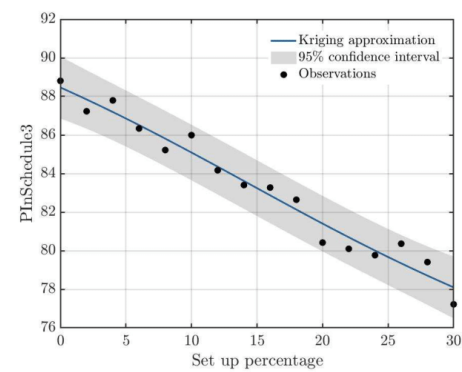
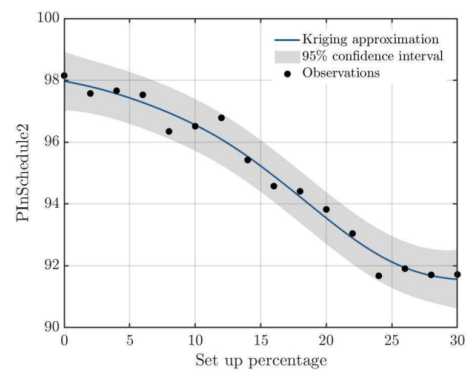
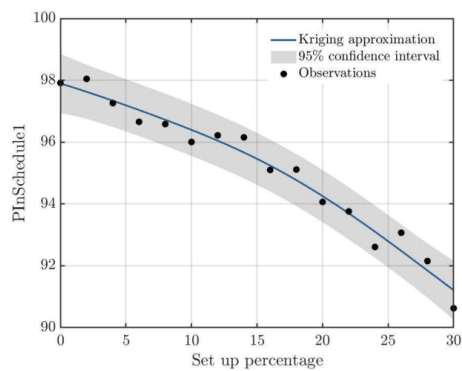
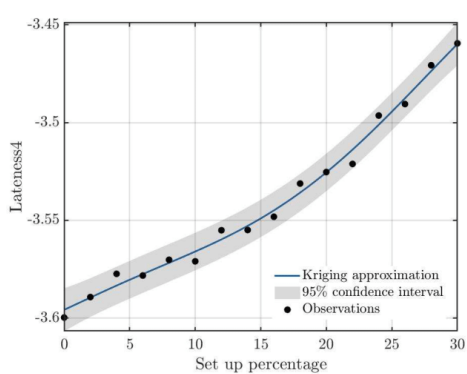
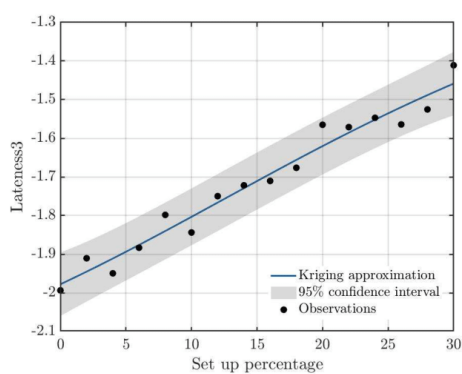
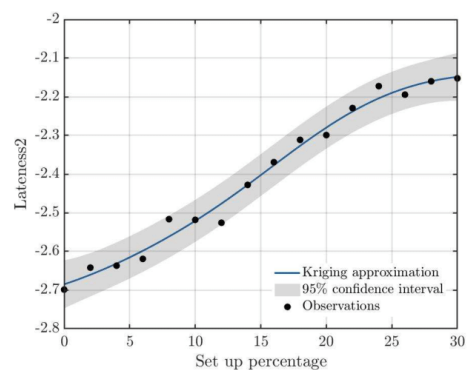
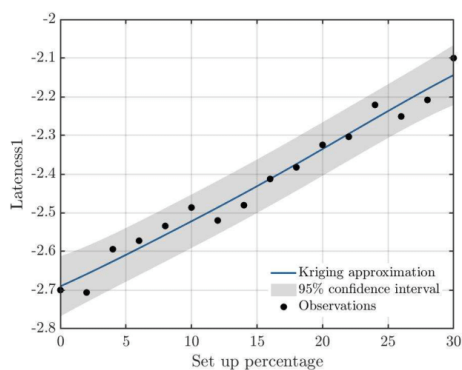


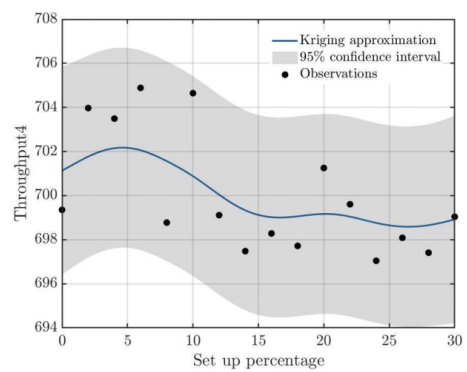
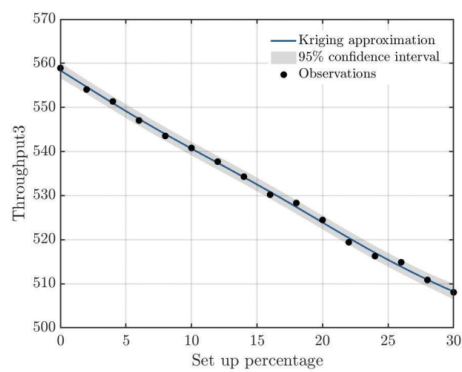
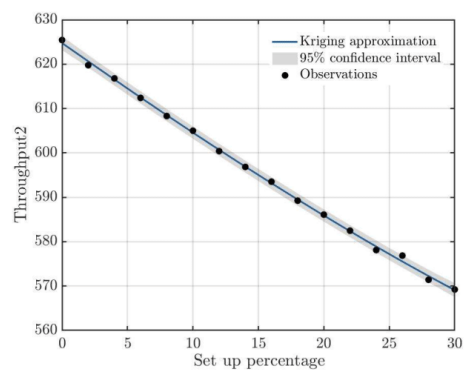
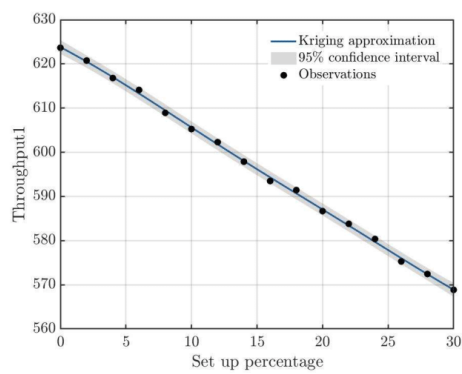
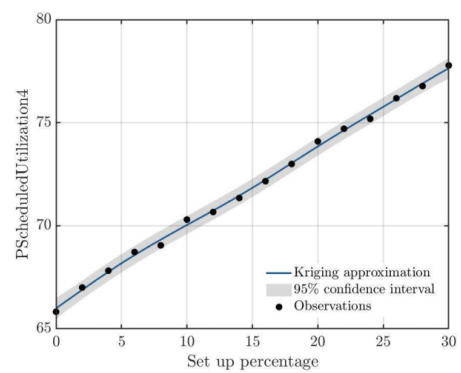
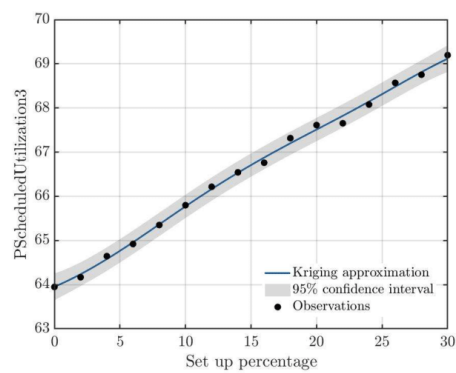
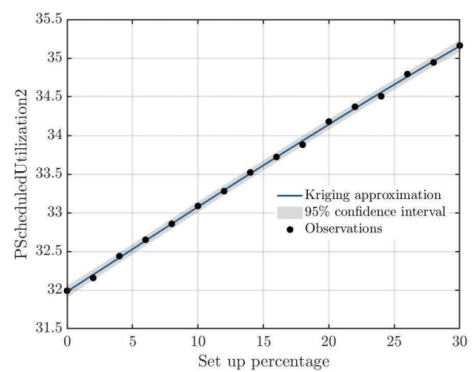
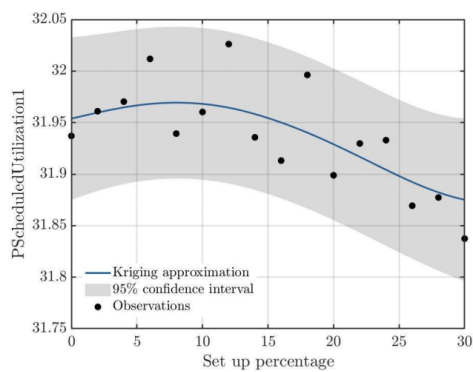




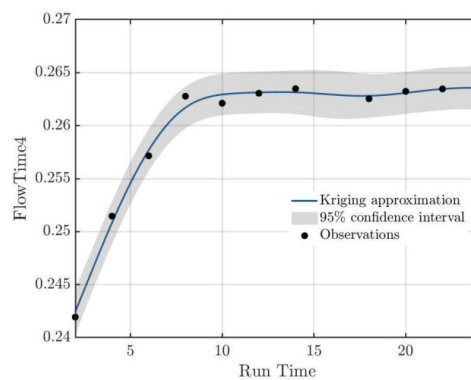
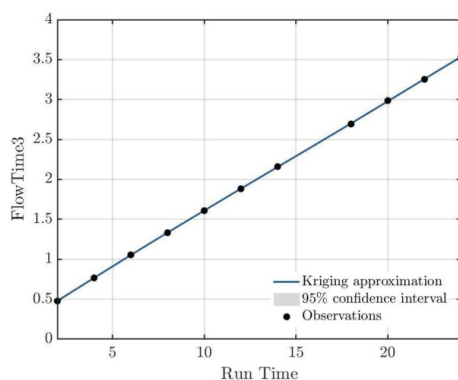
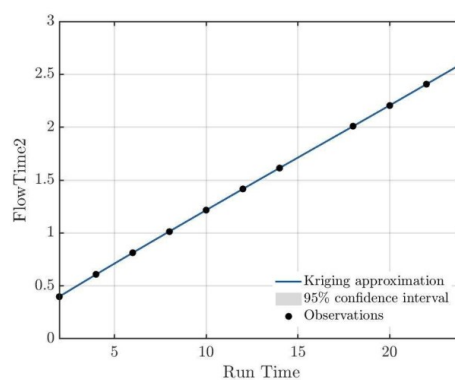
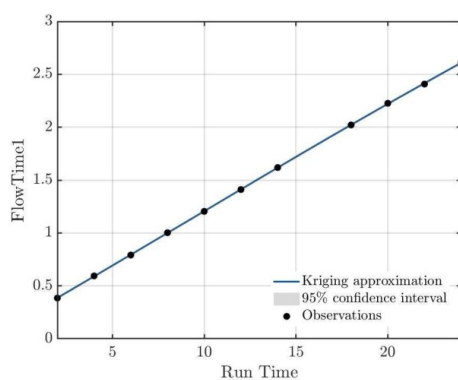
REGRA SHORTEST PROCESSING TIME COM 15 MINUTOS NA POOL

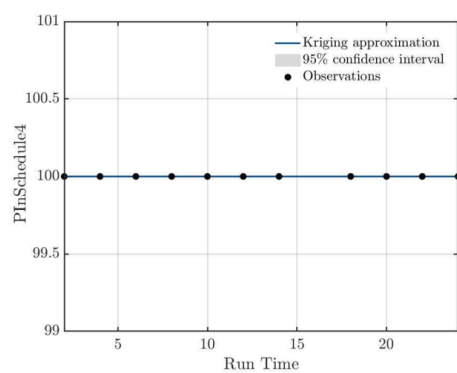
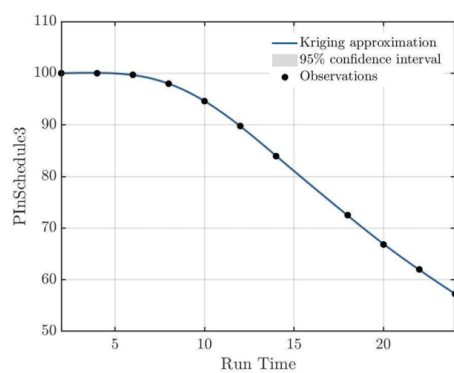
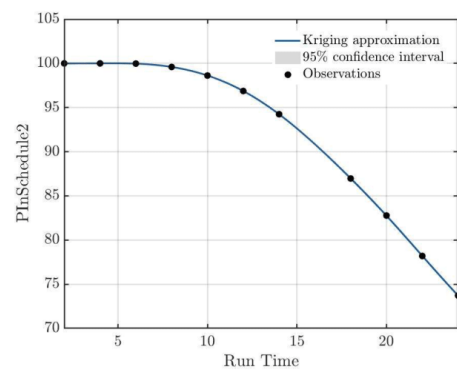
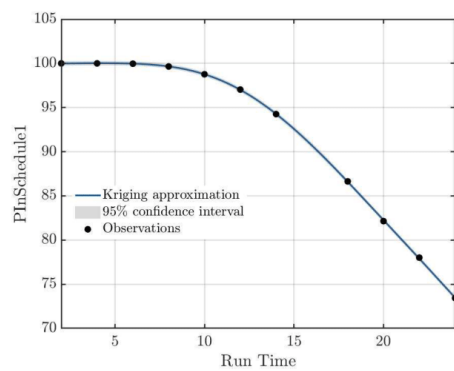
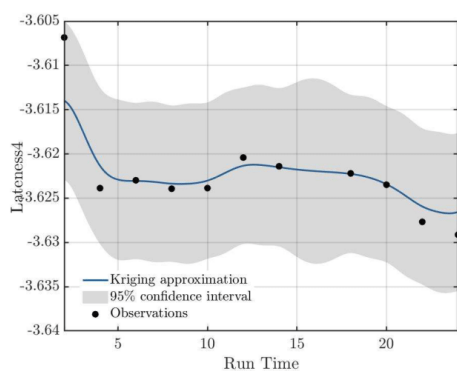
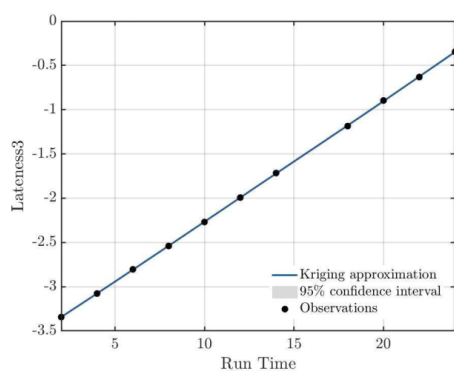
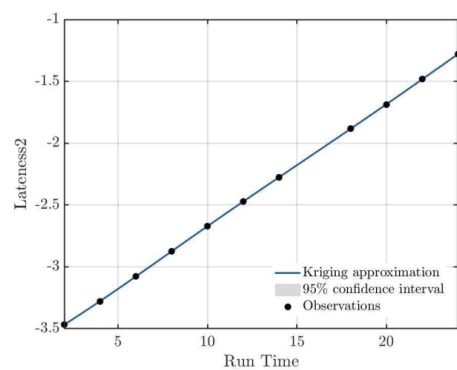
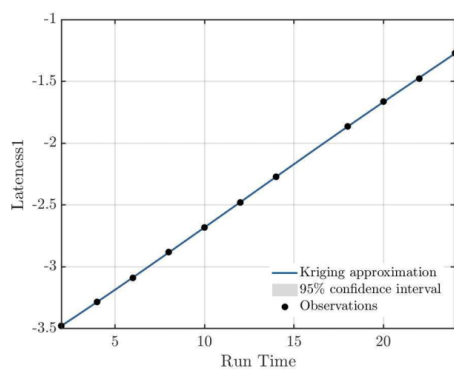


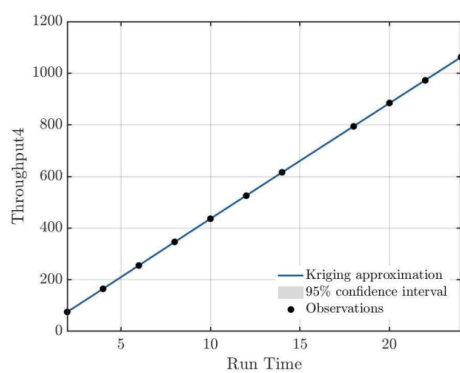
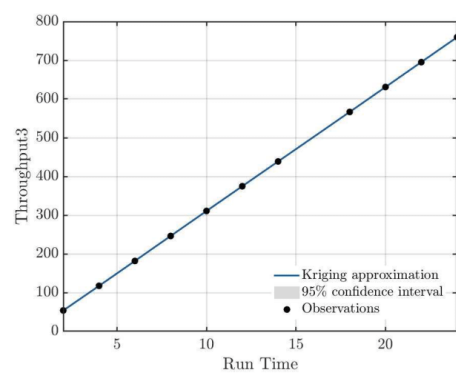
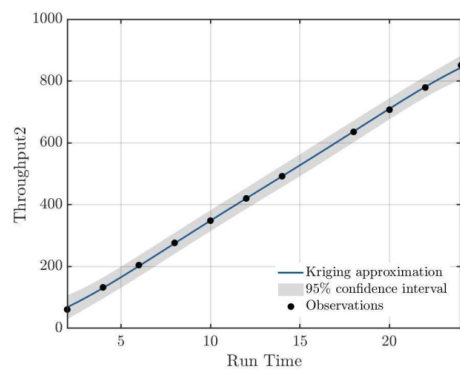
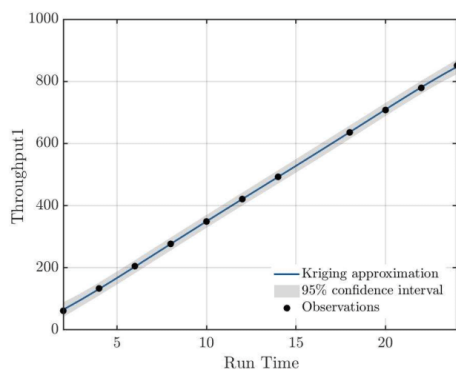
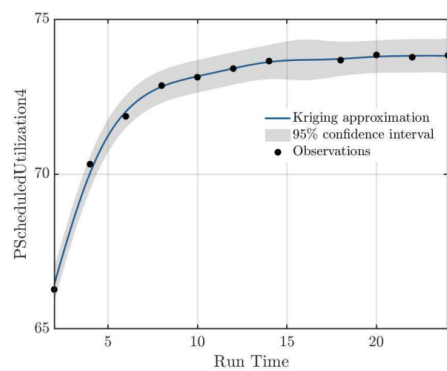
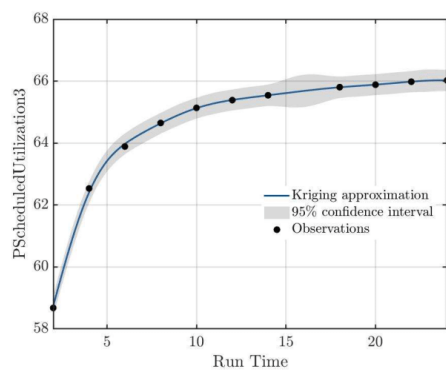
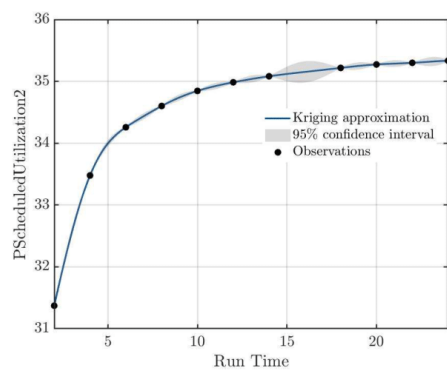
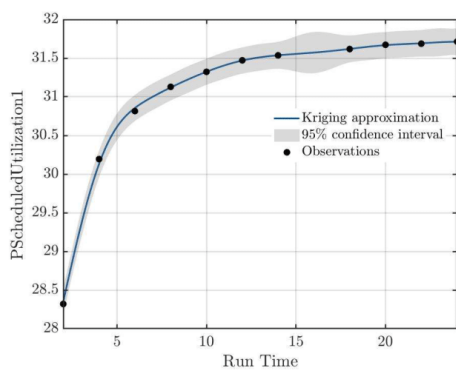




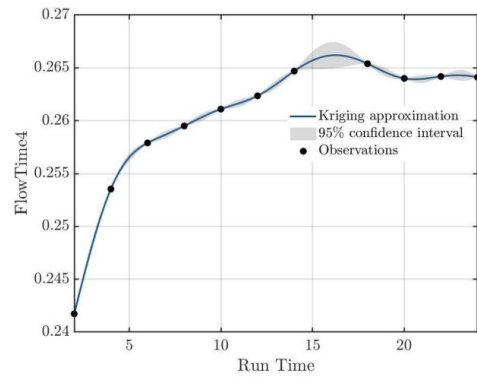
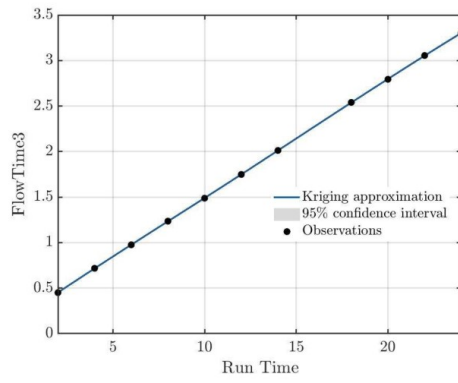
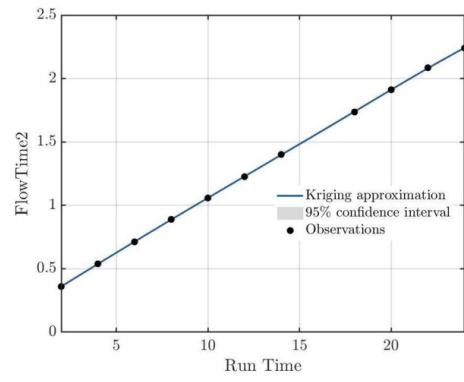
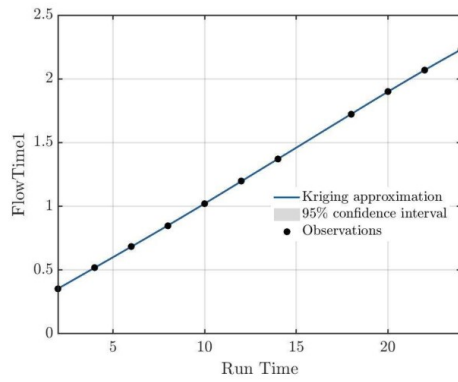
VARIAÇÃO DO TEMPO DE SIMULAÇÃO COM A REGRA EARLIEST DUE DATE

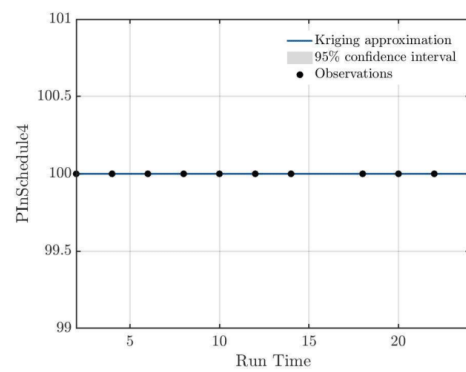
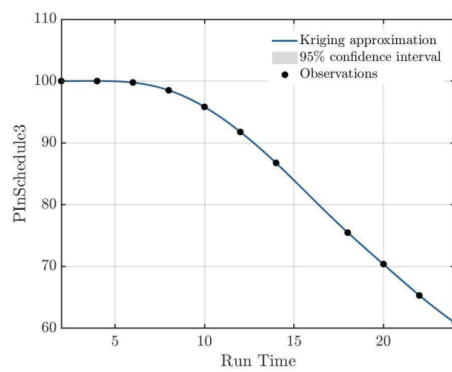
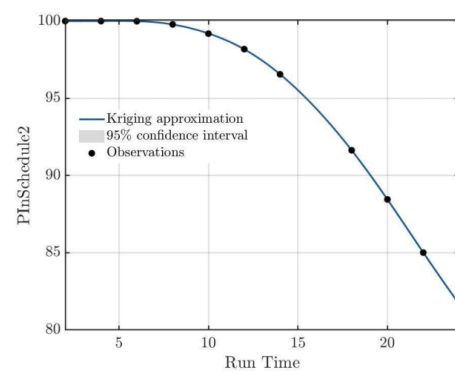
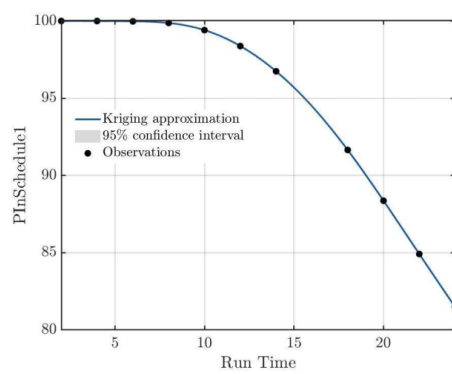
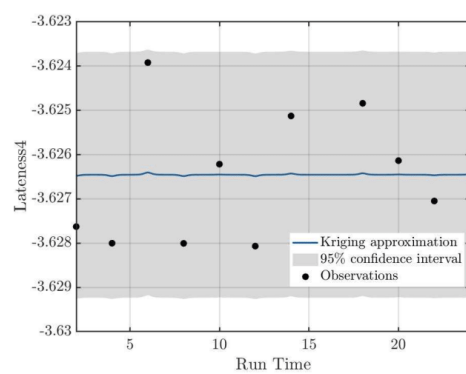
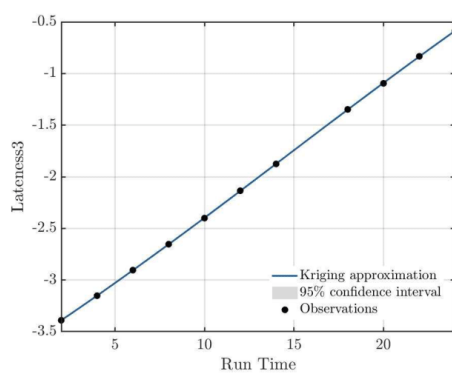
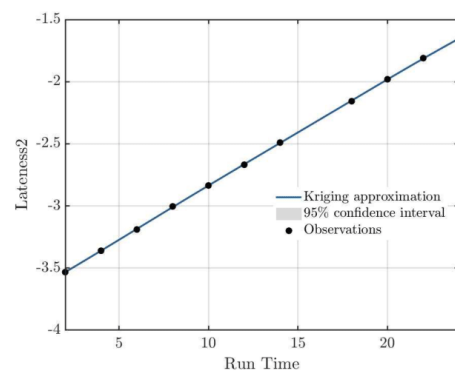
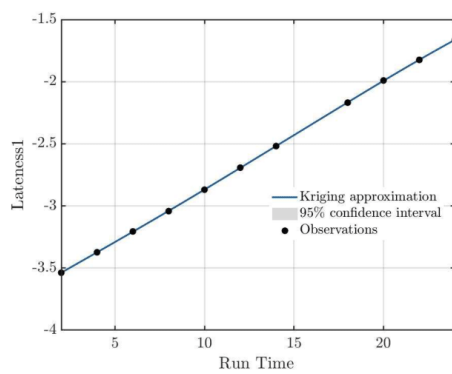


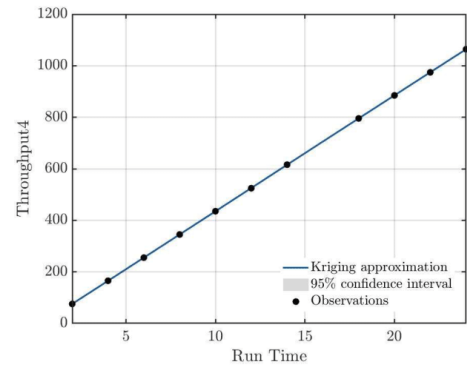
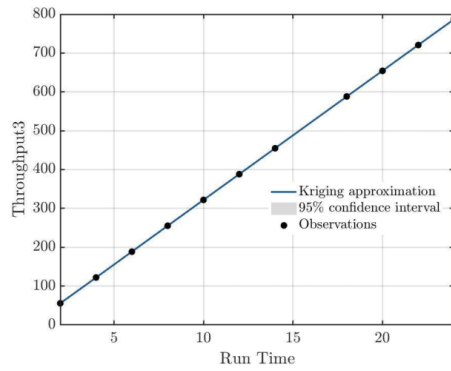
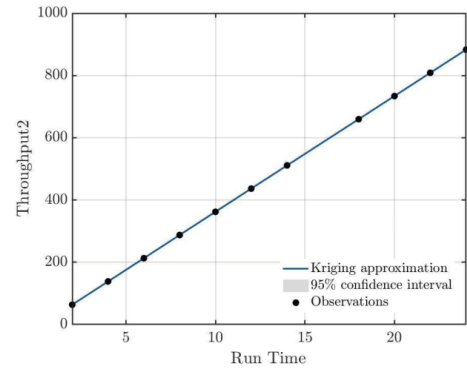
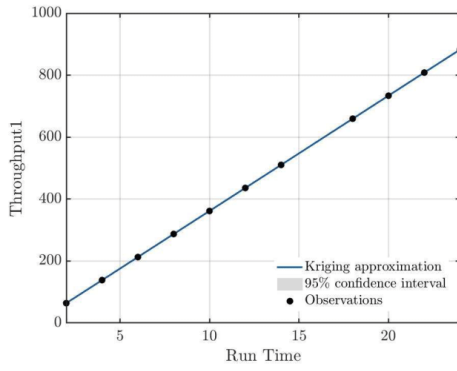
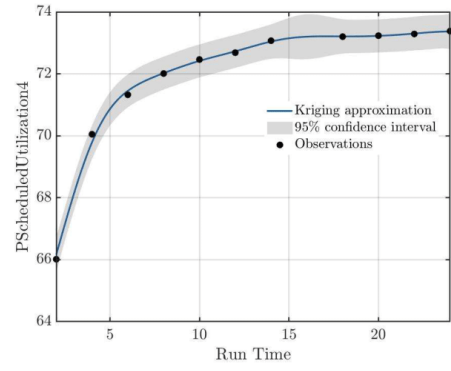
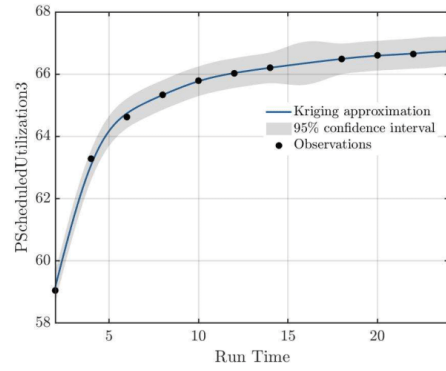
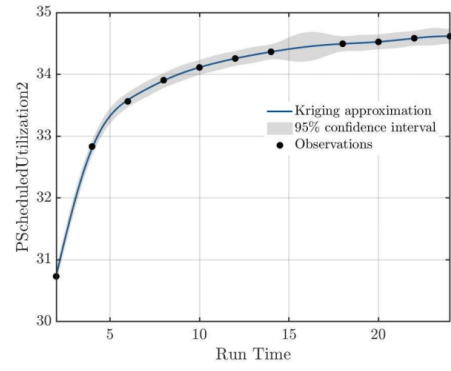
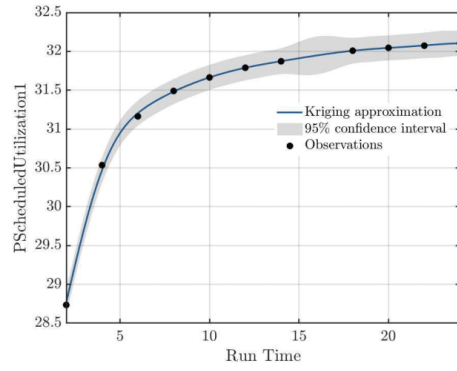




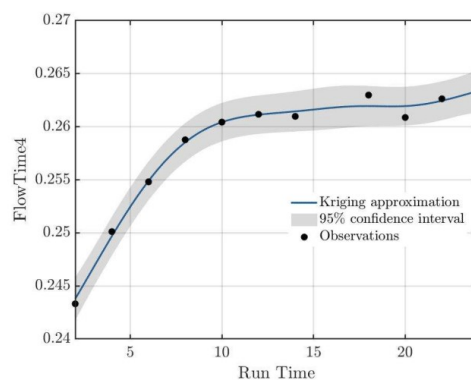
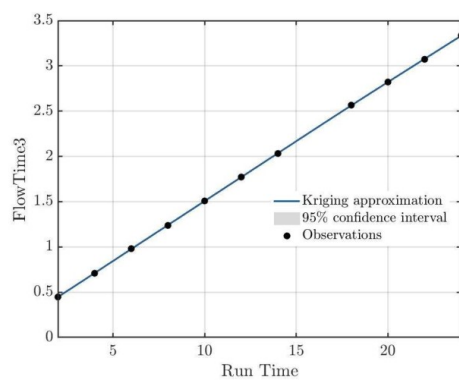
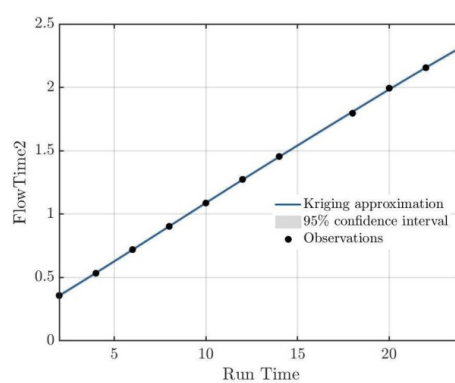
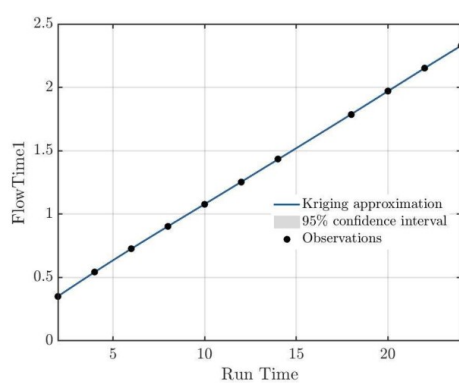
VARIAÇÃO DO TEMPO DE SIMULAÇÃO COM A REGRA
SHORTEST TOTAL PROCESSING TIME

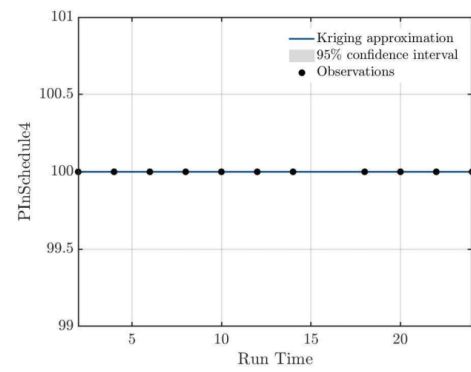
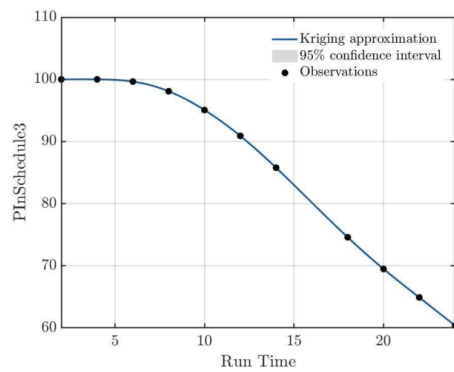
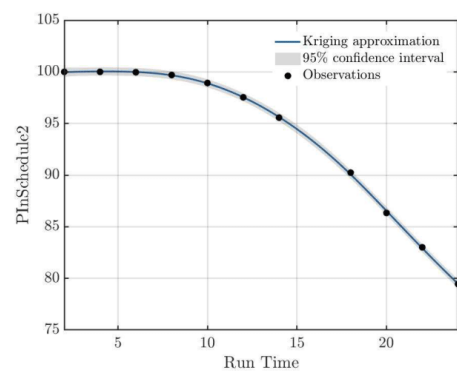
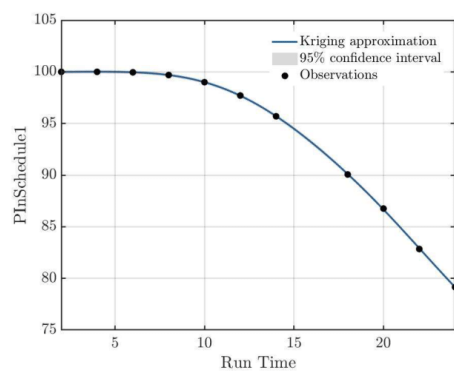
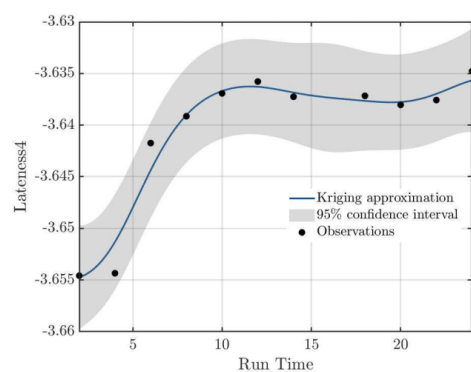
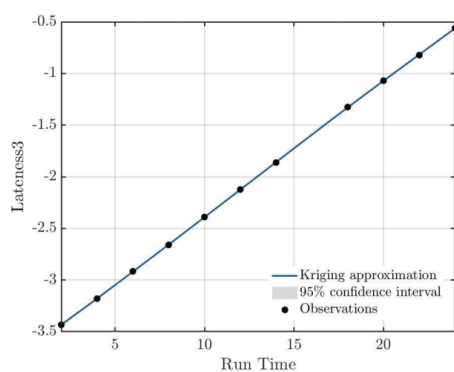
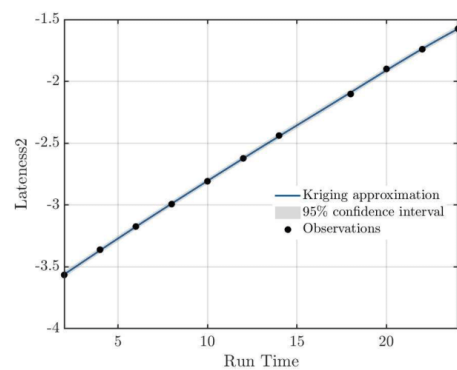
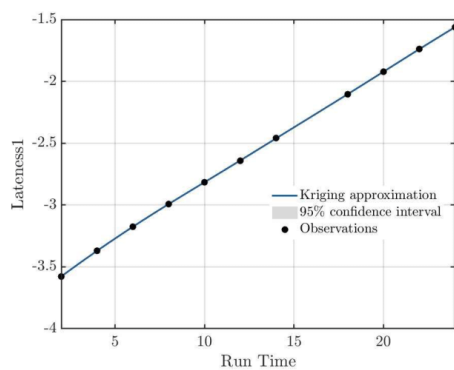


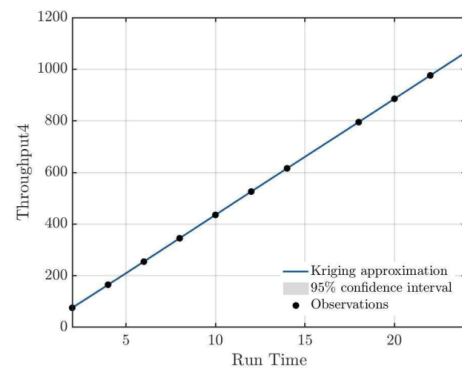
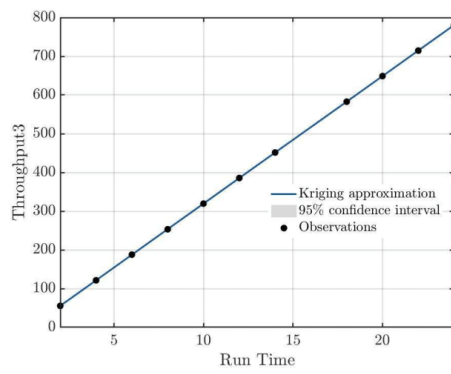
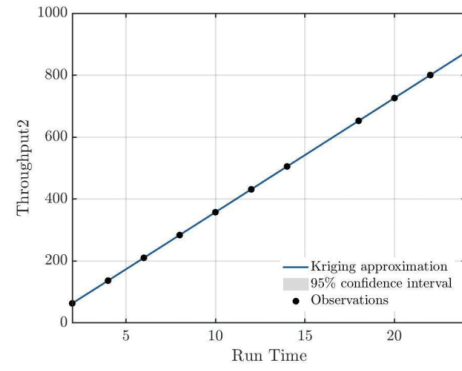
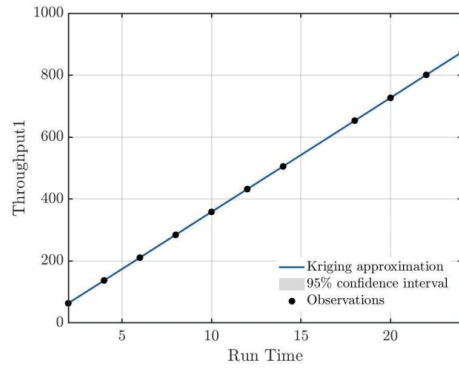
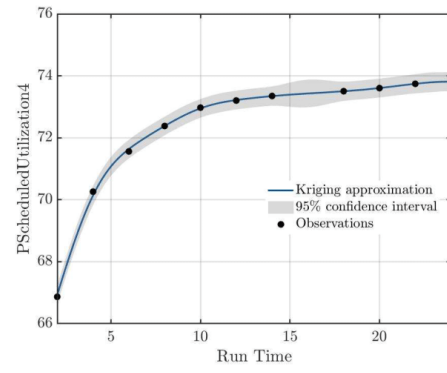
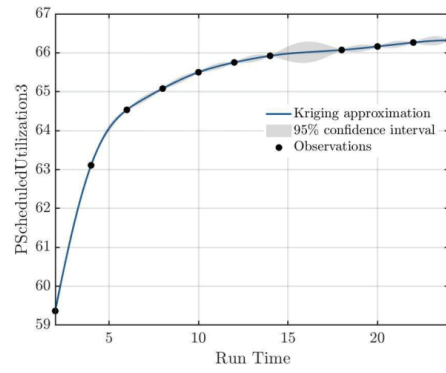
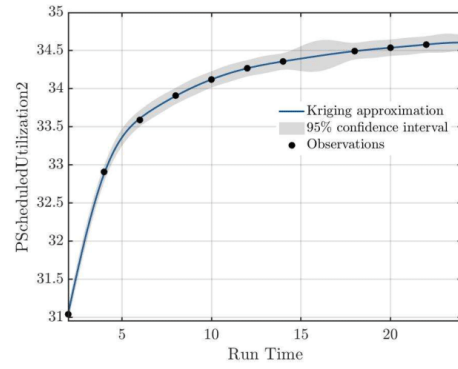
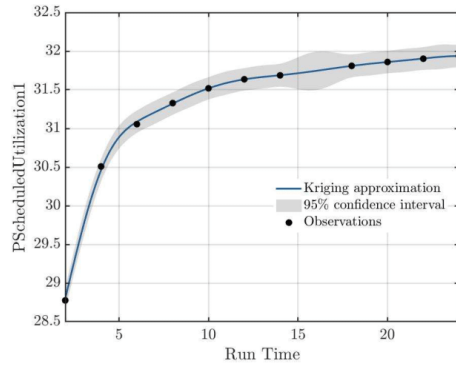




VARIAÇÃO DO TEMPO DE SIMULAÇÃO COM A REGRA LONGEST PROCESS SEQUENCE







VARIAÇÃO DO TEMPO DE SIMULAÇÃO COM A REGRA SHORTEST PROCESSING TIME

